

Homecomputer

5 1. Jahrgang

Mai '83 5,50 DM 45 öS 6,00 sfr

In diesem Heft u.a.:

VC-20

Editor
Assembler
HC-Invaders
Car-Crash

Sharp MZ-80K

Eidechse

VC-64

Geisterjäger
Mondlandung
Wurm
Sprite-Generator

ZX-81

Ritter
Missile ZX-Command

Apple II

Disk-Schutz 2

CBM

Mastermind (Computerhirn)
Mastermind II

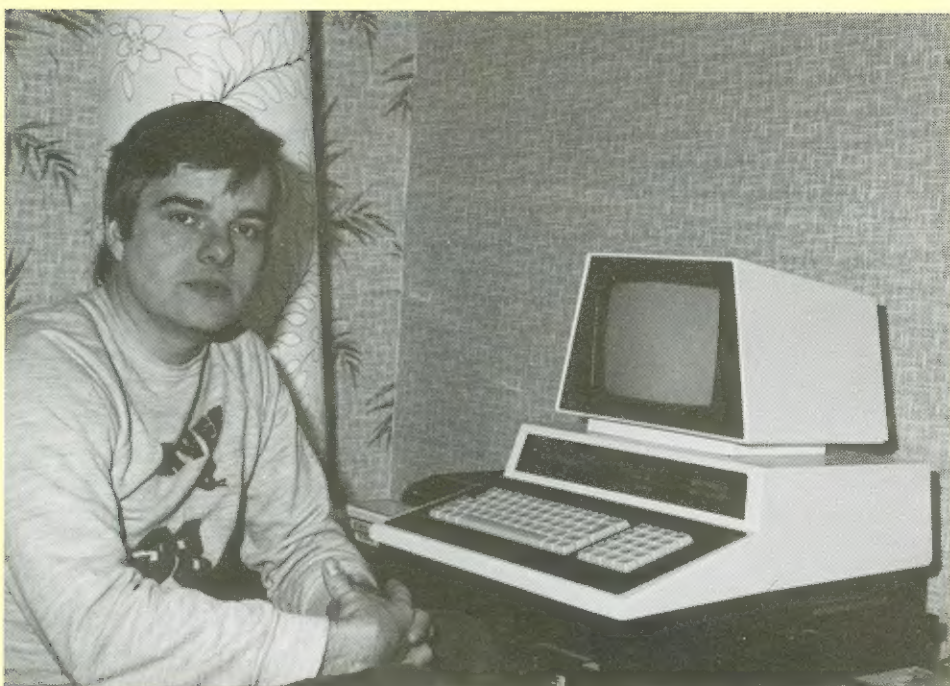
Texas-Special

mit vielen Programmen für den TI-99/4A

Jeden Monat mehr als
15 neue Programme
+ Tips + News
aus aller Welt

Große Werbeaktion!
250 Computer
zu gewinnen!





Liebe Leserin, lieber Leser,

wenn Sie diese Zeilen lesen, ist die Hannover-Messe 83 schon beendet.

Jetzt im Moment, in dem ich diese Zeilen schreibe, steht sie wenige Stunden bevor. 8 Tage lang werden wir uns bemühen, das Interessanteste für Sie, die Leser von Homecomputer, herauszupicken, werden wir versuchen, detaillierte Informationen über alles Neue und Berichtenswertes zusammenzufassen.

Im Juni-Heft werden wir dann darüber berichten. Darum wird die Neuheitenrubrik etwas vergrößert werden. Allerdings würde ein allumfassender Bericht den Rahmen von Homecomputer bei weitem sprengen und weniger veröffentlichte Software bedeuten.

Dies möchten wir nicht. Auch in Heft 6/83 sollen, wie bei uns üblich, mehr als 15 Programme enthalten sein.

Aus diesem Grunde bringen wir ein „Sonderheft Hannover-Messe“ heraus, damit Sie sich – auch falls Sie nicht die Gelegenheit hatten länger oder überhaupt diese wichtigste aller Messen zu besuchen – über alles ausführlich informieren können.

Microcomputer, Drucker, Datenspeichergeräte fürs Heim; darüber werden wir in Wort und Bild berichten und hoffen, Ihnen damit eine interessante Ergänzung zum gewohnten Homecomputer mit dem Programmservice zu bieten.

Herzlichst
Ihr Ralph Roeske

Herausgeber, Chefredakteur

Homecomputer erscheint monatlich im:
Roeske Verlag, Eschwege

Herausgeber:
Ralph Roeske

Redaktion:
Ralph Roeske (R),
Chefredakteur
(verantwortlich)
Cornelius Geppert (Ge)
Bärbel Voigt

Herstellung:
Roeske Verlag, Eschwege

Satz und Reproduktion:
Imtal Composing GmbH
3436 Hessisch Lichtenau

Druck:
Vogt GmbH
Druckerei und Verlag
3436 Hessisch Lichtenau

Vertrieb:
Inland (Groß-, Einzel- und
Bahnhofsbuchhandel), sowie
Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Str. 7
6200 Wiesbaden
Tel: 06121-2660

Anzeigenleitung:
Monika Roeske

Erscheinungsweise:
Erstverkaufstag von Home-
computer ist jeweils der letzte
Montag vor dem Erschei-
nungsmonat

Urheberrecht:
Alle in Homecomputer veröf-
fentlichen Beiträge sind urhe-
berrechtlich geschützt. Alle
Rechte, auch Übersetzungen,
vorbehalten. Reproduktionen
jeder Art (Fotokopie, Micro-
film, Erfassung in Datenver-
arbeitungsanlagen, usw.) be-
dürfen der schriftlichen Ge-
nehmigung des Verlages.
Alle veröffentlichte Software
wurde von Mitarbeitern des
Verlages oder von freien Mit-
arbeitern erstellt. Aus Ihrer
Veröffentlichung kann nicht
geschlossen werden, daß die
beschriebenen Lösungen
oder Bezeichnungen frei von
Schutzrechten sind.

Anschrift:
Homecomputer
Gartenstraße 47
3440 Eschwege
Tel: 05651-5993

Bezugspreise:
Einzelheft: 5,50 DM
Abonnement: Inland 55,- DM
pro Jahr (12 Ausgaben)
Ausland: Europa 80,- DM
USA 110,- DM

Anzeigenpreise:
Es gilt die Anzeigenpreisliste
Nr. 1 vom 1. Januar 1983
1/1 Seite S/W 3000,- DM
Farbzuschlag je Farbe
(Europaskala) 500,- DM

Nachdruck gestattet:
Auszüge von Homecomputer
(weniger als 10% des Redak-
tionsanteils) dürfen ohne Ge-
nehmigung in Schülerzeit-
schriften, Computer-Club
Magazinen und ähnlichen
nichtgewerblichen Publika-
tionen abgedruckt werden.

Bedingungen:
Es darf nur Originalmaterial
kopiert werden (also keine
Kopie einer Kopie). Jeder
Nachdruck muß an seinem
Beginn folgende Copyright-
Notiz in Original-Größe bein-
halten:

Copyright 1983
Homecomputer,
Gartenstraße 47
3440 Eschwege, BRD
Probeexemplar 5,50 DM
Jahresabonnement 55,- DM
Wir bitten um freie Zusen-
dung von 2 kostenlosen
Exemplaren jeder Publika-
tion, die nachgedrucktes Ma-
terial enthält.

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröf-
fentlichung gerne entgegen.
Honorare nach Vereinbarung.
Bei Zusendung von Manuskripten und Software erteilt
der Autor dem Verlag die Genehmigung zum Abdruck und
Versand der veröffentlichten Programme auf Datenträger.
Rücksendung erfolgt nur bei angeforderten Beiträgen,
ansonsten nur gegen Erstattung der Unkosten.

Zusendungen von Software zur Veröffentlichung sollte
bitte folgendes enthalten:
Kopierfähige Kassette oder Diskette mit dem Programm,
von Drucker erstelltes Listing oder Serie von Bildschirm-
fotos (keine Schreibmaschinen-Listings), eventuell Bild-
schirmfotos von einem Probe-
lauf.

Homecomputer bringt in diesem Monat:

Leserbriefe	2
News & Trends	3
Texas 99	
Panzerkrieg	5
Panzer bahnen sich den Weg durch unwegsames Gelände.	
TI-UFO	8
Ein Raumflug mit wachsendem Schwierigkeitsgrad	
Killersatellit	9
Der TI-99 versucht den Space-Shuttle des Spielers zu vernichten.	
Sharp MZ-80K	
Eidechse	10
Ein lustiges Spiel. Eine Eidechse muß so viele Fliegen wie mögl. fangen.	
Lesersoftware	
Chiffrier-/Dechiffrier-Programm	12
Der TI-99 als Ver- und Entschlüssler von geheimen Nachrichten.	
Rangliste	13
Der TI-99 sortiert erreichte Punktzahlen bei Sport und Spiel.	
Starship VC-20	
HC-Invaders	14
Ein Basic/Maschinenspracheprogramm für den 16K-VC-20	
Editor für 6502-Systeme	25
Fortsetzung der Reihe "Programmieren in Assembler"	
Labyrinth in 3.5 K	30
Es geht doch! Labyrinth im VC-20 ohne Erweiterung!	
Car-Crash	35
Ein Arkadenspiel bei dem man höllisch schnell schalten muß!	
Software	
Grandmaster - ein Meister seines Fachs.	34
Wir stellen ein interessantes Schachprogramm vor.	
Die ZX-Seiten	
Ritter	37
Ein Ritter (bewegt vom Spieler) versucht, in eine Burg zu gelangen und wird dabei von Pechwolken und wehrhaften Burgverteidigern bedrängt.	
Missile ZX-Command	41
Noch ein Arkadenspiel - die bekannte Städteverteidigung!	
VC-64 VideoChips	
Mondlandung VC-64	44
Ein Programm das es in sich hat. Nicht leicht, den "Adler" zu landen.	
Wurm	46
Ein Wurm kämpft ums Überleben.	
Geisterjäger	48
Ein hübsches Spiel mit der VC-64-Sprite-Grafik.	
VC-64-Sprites-Generator	50
Ein Programm zum einfachen Erzeugen und Speichern von Sprites.	
Apple-Kiste	
Diskschutz für Apple II	52
Disketten vor unbefugten Anwendern schützen.	
Trollhöhlen	55
Mit gefährlichen Bandwürmern, Trollen und sonstigen Ungeheuern.	
Pet-Bytes	
Computerhirn = Mastermind	57
Mastermind 2	59
Zwei Denkspele - einmal anders.	
PEEKs + POKEs	
Der VC-64-Videochip	62
Serienstart. Blicke in die Bytes des VC-64.	
Korrekturen	64
Software-Service	64

Heft 3 noch lieferbar

Ich bin angenehm beeindruckt von dem Inhalt Ihrer Zeitschrift Homecomputer. Leider habe ich das erste Heft nicht mehr am Kiosk bekommen. Bitte senden Sie mir das Heft 3/83 und die VC20-Kassette Heft 4.

P. Hirsch Köln
Es ist ab Mai noch eine begrenzte Zahl von Heft 3 zu haben.
Bitte bestellen Sie über den Verlag.

zu Slalom Heft 3/83

Zu Heft 3 habe ich folgende Frage:
Wie steuert man im Programm Slalom den Fahrer?

Johann Breitenfellner Linz
Die Tasten Z und / steuern den Alpinisten. Das haben wir vergessen abzdrukken.

ZX-Drucke

Mit großem Interesse habe ich Ihre Zeitschrift "Homecomputer" gelesen.

Beim Labyrinth-Programm für den ZX81 treten bei mir jedoch Schwierigkeiten auf. Das Unterprogramm in Maschinensprache, welches in einem Stringausdruck untergebracht ist, ist leider kaum lesbar.

Die Zeichen 8 und B sind teilweise nicht zu unterscheiden. Aus diesem Grund läuft das Unterprogramm bei mir nicht.

Mein Tip für zukünftige Ausgaben: Geben Sie das Programm in Maschinensprache in Dezimal an (wie bei U-Boot-Jagd), da dort eine Verwechslung zwischen Bund 8 nicht auftreten kann.

Thomas Kaus Burladingen 1

Negativ...

Positiv finde ich ...

Negativ fällt es daher auf, daß Eure Zeitschrift eben gar nicht niedriger im Preis rangiert. Sie ist sogar, vergleicht man Preis und Umfang, so ziemlich die teuerste deutsche Zeitschrift ihrer Art! (z.B. kostet Computer persönlich nur 4,- DM!)

Anzeigen in denen sich der Leser über den deutschen Geräte- und Softwaremarkt informieren kann, suchte ich in Homecomputer so gut wie vergeblich. Das ist nicht nur eine inhaltliche Lücke (ich kaufe eine Fachzeitschrift gerade auch wegen der Anzeigen!) - es heißt auch, daß der Käufer die Herstellungskosten alleine zu decken hat!

Franz Nagel Darmstadt

Besitzen Sie einen Homecomputer?

Sicher nicht, sonst hätten Sie das Preis/Leistungsverhältnis (von dem in der Computerbranche oft die Rede ist) schon er-

kannt. Für einen VC-20-Anwender z.B. sind allein die Programme der ersten drei Ausgaben von Homecomputer einige Jahresabos wert - und das bei nur 16.50 DM!

Nicht vollständig?

Wäre es nicht möglich, in zukünftigen Ausgaben alle (auch öde Tippereien wie Apple-Steno) Programme vollständig abzdrukken, da 1 Disk/Monat die Taschengeldgrenze manches Kleincomputeraners haushoch übersteigt.

Wolfgang Schmidt München

Teilen Sie uns bitte mit, welche Seite in Ihrem Heft gefehlt hat. Oder haben Sie etwa nur den Absatz zum Leser-Service auf Seite 33 mißverstanden?

Umsonst geärgert - Handbuch lesen.

Die Märzangabe von Homecomputer hat mich 505,50 DM gekostet! 5,50 für das Heft und für 500,00 DM habe ich mich geärgert. Ich habe nämlich das Robotprogramm (fehlerfrei!) eingetippt und dazu brauche ich ja anscheinend keine 16K-Erweiterung. Trotzdem läuft das Programm nicht. Nachdem die Leertaste gedrückt wurde ist es aus und vorbei, Errors, so weit das Auge reicht. Da ich fehlerfrei getippt habe (das habe ich bis zur "Vergasung" nachgeprüft), glaube ich nicht, daß der Fehler bei mir liegt.

Thomas Schiefer Eggenfelden

Und doch ist es so. Weil die meisten VC-20-Besitzer eine Speichererweiterung um mindestens 8 K besitzen, schreiben wir Programme fast immer für 8- oder 16-K-Versionen.

Daß dann die Adressen für Video- und Farbmatrix angepaßt werden müssen, wenn man weniger als 11.5 K freien Speicher hat, ist aus dem Handbuch zu ersehen.

Der VC-20 ist aber glücklicherweise der einzige Microcomputer, der uns auf diese Art und Weise angeschmiert hat, denn bei anderen Computern stellt sich das Problem nicht.

MID\$ mit Dreien?

Bei Ihrem Kalender-Programm für den PET bin ich mit Ihrer Zeile 135 nicht ganz einverstanden. MID\$ hat doch 3 Argumente. Ich kann leider aus Zeitgründen nicht das ganze Programm durchgehen und möchte Sie bitten den Fehler zu korrigieren.

Stefan Munch Erlensee

Gibt es etwas zu korrigieren? Wer das Computerhobby ernsthaft betreibt, muß sich hin und wieder ein wenig Zeit nehmen, zu lernen, was noch nicht gelernt oder verstanden wurde. Probieren Sie den MID\$ mit 2 Argumenten aus. Sie werden überrascht sein.

Ferien im Computer-Camp

Super Freizeitpaß mit Computer-learning für junge Leute!

Im Luftkurort "Haltern am See" findet am 310 ha großen Trinkwassersee der Stevertalsperre eines der ersten europäischen Computercamps statt. Junge Leute werden hier spielerisch mit dem Mikrocomputer und der Programmiersprache BASIC vertraut gemacht. Durch intensive Beschäftigung mit dem Mikrocomputer

COMMODE (für je zwei Teilnehmer steht eine Commodore-Anlage zur Verfügung) wird die Angschwelle schnell abgebaut und der Umgang mit der EDV-TECHNIK zur Selbstverständlichkeit. Vorgesehen sind: 50 Unterrichtsstunden Computerlearning unter Anleitung von Fachdozenten, umfangreiches Unterrichtsmaterial, Einsatz von Video, zusätzlicher theore-

tischer Unterricht, Einführung in die EDV und Mikrocomputer, sowie deren Möglichkeiten und Grenzen. Neben dem Computerlearning steht der Kontakt der Teens und Twens mit Sport, Spiel und Freizeitpaß im Vordergrund. Das Freizeit-Aktionsprogramm bietet (teilweise gegen geringe Unkostenbeteiligung) Fußball, Schwimmen, Reiten, Fahrradverleih u. a.. Für das Rundumprogramm sind weiterhin vorgesehen: Planwagenfahrten, Nachtwanderungen, Lagerfeuer und heiße Disconächte. Die Unterbringung, die gemeinsamen Mahlzeiten, die Programmierung der Mikrocomputer COMMODE findet in der "Jugendherberge Haltenersausee" Overath 39, 4358 Haltern-Stausee, statt.

111 Cassetten im Programm

**BUG-BYTE
SOFTWARE** 
AUTHORISED DEALER

Software-Anbieter für die populären Homecomputer sind zur Zeit noch nicht sehr zahlreich gesät. Einer der größten Anbieter für ZX 81, Spectrum, VC 20, Acorn, BBC und Dragon ist der Softwareversand Augsburg, der jetzt auch den Spectrum-Computer im Programm hat. Produkte der Firmen Artic, Bug Byte, Melbourne House, Publishers Personal Soft-

ware A+F und Impact werden zum Teil exklusiv vertrieben.

Alles mit ausführlicher (allerdings englischer) Beschreibung und original verpackt. Ein Katalog mit dem vorhandenen Programmspektrum ist gegen Gebühr erhältlich.

Softwareversand
Thomas Wagner, Augsburg

Microdrive bald lieferbar.

Noch immer wartet die Homecomputerwelt auf die Fertigstellung des vor 1½ Jahren angekündigten Microdrives.

Laut Sinclair stehen lediglich noch die speziell für die Firma hergestellten Halbleiter-Chips aus, die zur Zeit noch eine Testphase durchlaufen. Das Design steht ebenfalls und in Kürze sollen für £40 per Stück die ersten 100.000 Geräte zunächst an die ersten Spectrum-Kunden geliefert und dann über den Handel auch der Rest der Interessenten bedient werden.

Somit wird dann auch ein Gerücht sein Ende finden,

nämlich, ob es sich um Microflopies oder um "Microbänder", ähnlich den sogenannten "Stringy Floppies", handelt. Laut unseren letzten vorliegenden Informationen handelt es sich um ein Endlosband, das sich mit hoher Geschwindigkeit innerhalb 7 Sekunden einmal am Übertragungskopf vorbeibewegt. So wird eine durchschnittliche Zugriffszeit von 3,5 Sekunden erreicht.

TRS-80/VG-Club in Wuppertal

Zu unserem Aufruf anlässlich der Clubecke im letzten Heft sandte uns Herr Gerd Breidenbruch aus Wuppertal folgende Mitteilung.

Der TRS-80/Mod.1 hat meines Erachtens einen sehr großen Marktanteil und ist immerhin der 'Homecomputer mit dem größten Software-Angebot'

Vielleicht interessiert es Sie in diesem Zusammenhang, daß es in Wuppertal einen TRS-80/Video-Genie Club gibt (Anzahl der Mitglieder zur Zeit ca. 40, eigene Clubzeitschrift, Clubbeitrag keiner).

Die Mitglieder des TRS-80/VG Clubs treffen sich monatlich einmal in der Universität Wuppertal Raum 113.34. Jeder Computer-Fan ist dort herzlich willkommen.

Da die Treffen noch in unregelmäßigen Abständen stattfinden, sollten sich Interessenten erst bei dem Leiter der Gruppe, Herrn Horst Brodowski, Gördelestr. 7, 5657 Haan/Rhl. Tel.: 02129/4891 erkundigen.

Wir bedanken uns für die freundliche Unterstützung bei der Titelgestaltung beim

Round up Western Shop

Frankfurter Straße 88 · 3500 Kassel

ORANGE 2 COMPUTER (voll APPLE kompatibel)

Groß-/Kleinschreibung, Umlaute bei allen Ausführungen

ARHELGER
electronic



Komplettsystem wie abgebildet

Bausatz: Hauptplatine 48K-RAM und alle Teile, fertiges Netzteil, Cherry-Tastatur (4 Cursortasten, 2 Editiertasten, Hometaste, Gehäuse)

Fertiggerät: wie oben, jedoch fertig aufgebaut,

Orange 2 6 Monate Garantie
Gehäuse für Rechner und 2 Drives auf Anfrage.

Profitaustatur zur Textverarbeitung:

Zehnerblock, Funktionstasten, Cursortasten, Umlaute, anschlußfertig im Gehäuse.

PERIPHERIE:

NEC-8023 BC m. Graphikinterf.

BMC-Monitor Typ 12 grün geätzte Röhre

Zenit-Monitor ZUM 121 (LOW COST)

16K Language Card

80 Zeichenkarte m. 40/80 Z. Softswitch

CP/M Z-80 CPU-Karte

Verbatim Datalife Disketten 5¼" (10)

Diskettenlaufwerke und weiteres Zubehör auf Anfrage!

Alle Preise verstehen sich inkl. 13% MwSt.
Weitere Informationen gegen DM 1,50 in BM.

ab DM **4499,-**

DM **1388,-**

nur DM **1688,-**

Aufpreis nur DM **488,-**

DM **1950,-**

DM **540,-**

DM **305,-**

DM **220,-**

DM **495,-**

DM **375,-**

DM **79,-**

Versand erfolgt unfrei.

3544 WALDECK-SACHSENHAUSEN
TELEFON 05634-1724

WINTERHAGEN 2
991160 AEW D

Panzerkrieg

TI-99/4A + Fernbedienungen

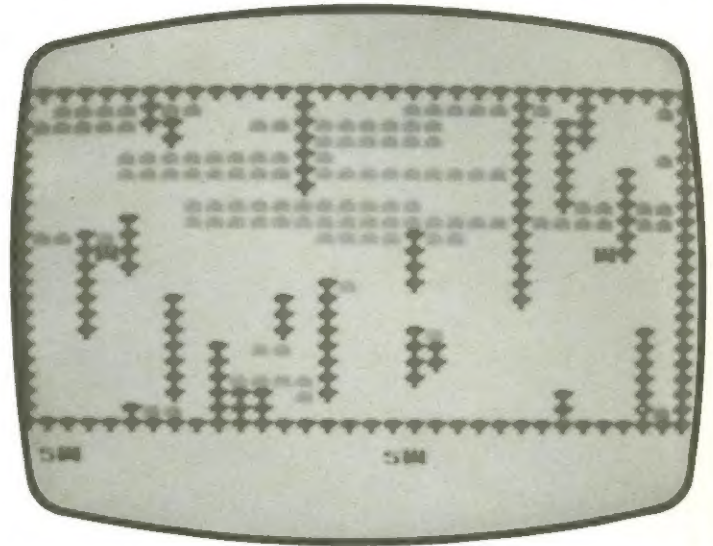
Der Sinn dieses Zwei-Personen-Spieles besteht darin, die gegnerischen Panzer abzuschießen und dabei möglichst wenig Flur- und Sachschaden anzurichten. Am Anfang stehen sich die Feinde gegenüber, getrennt durch Baum- und Häuserreihen. Die Häuser können zwar abgeschossen werden, bringen aber 300 Minuspunkte ein. Will man ein Haus umfahren,

wird der Panzer zerstört und der Gegner bekommt 1000 Pluspunkte. Ebenso bei Abschluß. Die Bäume können weder abgeschossen noch umgefahren werden, "wehren" sich jedoch mit Minuspunkten.

Anm.: Die Zeilen 740-930 verbinden, daß ein Panzer am Anfang von Dämmen umgeben und bewegungsunfähig ist. Sollten die Gegner jedoch durch eine

durchgehende Baumreihe getrennt sein, muß das Spiel mit "CLEAR" abge-

brochen und mit "RUN" neu gestartet werden.
Jürgen Hoppstädter



```

1000 ST 210 DIMIZE
1010 I=1 TO 30 ZEICHEN DEFINIEREN
1020 CALL CHAR(128,"DBDBDBFF")
1030 CALL CHAR(129,"FFFF787F")
1040 CALL CHAR(130,"C3FFFFFF")
1050 CALL CHAR(131,"FFFF1EF7")
1060 CALL CHAR(132,"00000181")
1070 CALL CHAR(133,"00003C24")
1080 CALL CHAR(134,"007E4242")
1090 CALL CHAR(135,"FF818181")
1100 CALL CHAR(136,"DBDBDBFF")
1110 CALL CHAR(137,"FFFF787F")

```

```

1120 CALL CHAR(138,"C3FFFFFF")
1130 CALL CHAR(139,"FFFF1EF7")
1140 CALL CHAR(140,"00000181")
1150 CALL CHAR(141,"00003C24")
1160 CALL CHAR(142,"007E4242")
1170 CALL CHAR(143,"FF818181")
1180 CALL CHAR(144,"3C7FFF7E")
1190 CALL CHAR(152,"00387CFE")
1200 CALL CHAR(153,"00003C24")
1210 CALL CHAR(154,"007E4242")
1220 CALL CHAR(155,"FF818181")
1230 REM FARBEN DEFINIEREN

```

```

1240 ST 340-550 SCREEN(15)
1250 CALL COLOR(13,5,1)
1260 CALL COLOR(14,14,1)
1270 CALL COLOR(15,13,1)
1280 CALL COLOR(16,10,1)
1290 CALL HOUSE BAUEN
1300 CLEAR
1310 I=1 TO 30
1320 RND+1
1330 RND+1
1340 HCHAR(X,Y,152,Z)
1350 BAEUME PFLANZEN
1360 I=1 TO 30

```

```

1370 X=23+RND+1
1380 Y=31+RND+1
1390 Z=7+RND+1
1400 CALL VCHAR(X,Y,144,Z)
1410 NEXT I

```

```

1420 ST 560-750 RAND/PUNKTETABELLE
1430 CALL HCHAR(24,1,32,32)
1440 CALL HCHAR(23,1,32,32)
1450 CALL HCHAR(22,1,32,32)
1460 CALL HCHAR(21,1,32,32)
1470 CALL HCHAR(20,1,32,32)
1480 CALL HCHAR(19,1,32,32)
1490 CALL HCHAR(18,1,32,32)
1500 CALL HCHAR(17,1,32,32)
1510 CALL HCHAR(16,1,32,32)
1520 CALL HCHAR(15,1,32,32)
1530 CALL HCHAR(14,1,32,32)
1540 CALL HCHAR(13,1,32,32)
1550 CALL HCHAR(12,1,32,32)
1560 CALL HCHAR(11,1,32,32)
1570 CALL HCHAR(10,1,32,32)
1580 CALL HCHAR(9,1,32,32)
1590 CALL HCHAR(8,1,32,32)
1600 CALL HCHAR(7,1,32,32)
1610 CALL HCHAR(6,1,32,32)
1620 CALL HCHAR(5,1,32,32)
1630 CALL HCHAR(4,1,32,32)
1640 CALL HCHAR(3,1,32,32)
1650 CALL HCHAR(2,1,32,32)
1660 CALL HCHAR(1,1,32,32)
1670 CALL HCHAR(0,1,32,32)
1680 CALL HCHAR(0,0,32,32)
1690 CALL HCHAR(0,0,32,32)
1700 CALL HCHAR(0,0,32,32)
1710 CALL HCHAR(0,0,32,32)
1720 CALL HCHAR(0,0,32,32)
1730 CALL HCHAR(0,0,32,32)
1740 CALL HCHAR(0,0,32,32)
1750 CALL HCHAR(0,0,32,32)
1760 CALL HCHAR(0,0,32,32)
1770 CALL HCHAR(0,0,32,32)
1780 CALL HCHAR(0,0,32,32)
1790 CALL HCHAR(0,0,32,32)
1800 CALL HCHAR(0,0,32,32)
1810 CALL HCHAR(0,0,32,32)
1820 CALL HCHAR(0,0,32,32)
1830 CALL HCHAR(0,0,32,32)
1840 CALL HCHAR(0,0,32,32)
1850 CALL HCHAR(0,0,32,32)
1860 CALL HCHAR(0,0,32,32)
1870 CALL HCHAR(0,0,32,32)
1880 CALL HCHAR(0,0,32,32)
1890 CALL HCHAR(0,0,32,32)
1900 CALL HCHAR(0,0,32,32)
1910 CALL HCHAR(0,0,32,32)
1920 CALL HCHAR(0,0,32,32)
1930 CALL HCHAR(0,0,32,32)
1940 CALL HCHAR(0,0,32,32)
1950 CALL HCHAR(0,0,32,32)
1960 CALL HCHAR(0,0,32,32)
1970 CALL HCHAR(0,0,32,32)
1980 CALL HCHAR(0,0,32,32)
1990 CALL HCHAR(0,0,32,32)
2000 CALL HCHAR(0,0,32,32)

```

```

2010 CALL GCHAR(XP1-1,YP1,ZP1)
2020 IF ZP1<>144 THEN 840
2030 CALL GCHAR(XP1+1,YP1,ZP1)
2040 IF ZP1<>144 THEN 840
2050 CALL GCHAR(XP1,YP1-1,ZP1)
2060 IF ZP1<>144 THEN 840
2070 CALL GCHAR(XP1,YP1+1,ZP1)
2080 IF ZP1<>144 THEN 840
2090 CALL GCHAR(XP1,YP1,ZP1)
2100 IF ZP1<>144 THEN 840
2110 CALL GCHAR(XP1,YP1,ZP1)
2120 IF ZP1<>144 THEN 840
2130 CALL GCHAR(XP1,YP1,ZP1)
2140 IF ZP1<>144 THEN 840
2150 CALL GCHAR(XP1,YP1,ZP1)
2160 IF ZP1<>144 THEN 840
2170 CALL GCHAR(XP1,YP1,ZP1)
2180 IF ZP1<>144 THEN 840
2190 CALL GCHAR(XP1,YP1,ZP1)
2200 IF ZP1<>144 THEN 840
2210 CALL GCHAR(XP1,YP1,ZP1)
2220 IF ZP1<>144 THEN 840
2230 CALL GCHAR(XP1,YP1,ZP1)
2240 IF ZP1<>144 THEN 840
2250 CALL GCHAR(XP1,YP1,ZP1)
2260 IF ZP1<>144 THEN 840
2270 CALL GCHAR(XP1,YP1,ZP1)
2280 IF ZP1<>144 THEN 840
2290 CALL GCHAR(XP1,YP1,ZP1)
2300 IF ZP1<>144 THEN 840
2310 CALL GCHAR(XP1,YP1,ZP1)
2320 IF ZP1<>144 THEN 840
2330 CALL GCHAR(XP1,YP1,ZP1)
2340 IF ZP1<>144 THEN 840
2350 CALL GCHAR(XP1,YP1,ZP1)
2360 IF ZP1<>144 THEN 840
2370 CALL GCHAR(XP1,YP1,ZP1)
2380 IF ZP1<>144 THEN 840
2390 CALL GCHAR(XP1,YP1,ZP1)
2400 IF ZP1<>144 THEN 840
2410 CALL GCHAR(XP1,YP1,ZP1)
2420 IF ZP1<>144 THEN 840
2430 CALL GCHAR(XP1,YP1,ZP1)
2440 IF ZP1<>144 THEN 840
2450 CALL GCHAR(XP1,YP1,ZP1)
2460 IF ZP1<>144 THEN 840
2470 CALL GCHAR(XP1,YP1,ZP1)
2480 IF ZP1<>144 THEN 840
2490 CALL GCHAR(XP1,YP1,ZP1)
2500 IF ZP1<>144 THEN 840

```

```

2510 YP2=YP2-1
2520 GOTO 840
2530 CALL HCHAR(XP1,YP1,128)
2540 CALL HCHAR(XP2,YP2,136)
2550 REM SPIELHAUPTPROGRAMM
2560 CALL GCHAR(XP1,YP1,ZP1)
2570 CALL JOYST(1,X1,Y1)
2580 IF Y1<>4 THEN 1010
2590 GOSUB 1240

```


Texas 99

TI-UFO

Ziel des Spieles ist es, ein Ufo um Hindernisse zu lenken. Die horizontale Steuerung erfolgt mit den Tasten A (nach links) und L (nach rechts). Wird keine Taste gedrückt, so bewegt

sich das Ufo vertikal nach unten. Wird ein Hindernis senkrecht überfahren, so werden 2 Fehlerpunkte berechnet, beim waagrechten Überfahren 1 Fehlerpunkt. Ebenso werden beim waagrechten Zurückfahren und bei Erreichen des Bildschirmrandes Fehlerpunkte errechnet.

Die ersten drei Runden be-

stehen aus drei Durchgängen. Jede weitere Runde aus zwei Durchgängen. Der Schwierigkeitsgrad nimmt ständig zu. Werden in einer Runde weniger als 8 Fehlerpunkte verursacht, so kommt man in die nächste Runde (Freispiel). Der Schwierigkeitsgrad kann in Zeile 170, der Schwierigkeitszuwachs in der Zeile

770 verstellt werden.
Runden 1-3: # \$ %
Runden 4 - 8: A B C D E
Runden 9 - : Zufallszeichen
Nachteil des Programms: Das Ufo kann aus Zeitgründen nicht gelöscht werden. Es bildet sich deshalb eine Ufoschlange.

Hans-Peter Adrion



```

100 CALL SCREEN(14)
110 CALL COLOR(1,16,14)
120 BU=1
130 JU=0
140 FE=0
150 ZE=35
160 REM HINDERNISSE PRO ZEIL
E
170 PIT=5
180 SC=0
190 FOR T=3 TO BU STEP -1
200 IF FE>=8 THEN 1050
210 CALL CLEAR
220 G=0
230 RANDOMIZE
240 GOSUB 560
250 IF FE>=8 THEN 1050
260 CALL SOUND(400,131,2,165
,2,196,2)
270 REM **UFO SETZEN
280 SPAL=16
290 CALL CHAR(130,"000000183
C7E7E99")
300 G=G+1
310 CALL GCHAR(G,SPAL,WE)
320 IF WE>33 THEN 330 ELSE 3
60
330 CALL SOUND(100,-2,1)
340 CALL COLOR(1,5,14)
350 SC=SC+2
360 CALL HCHAR(G,SPAL,130)
370 IF G=24 THEN 660
380 CALL KEY(0,KEY,STATUS)
390 IF STATUS=0 THEN 300
400 IF KEY=65 THEN 460
410 IF KEY=76 THEN 430
420 GOTO 300
430 SPAL=SPAL+1
440 IF SPAL=33 THEN 460
450 GOTO 480

```

```

460 SPAL=SPAL-1
470 IF SPAL=0 THEN 430
480 CALL GCHAR(G,SPAL,Q)
490 IF Q>33 THEN 500 ELSE 53
0
500 CALL SOUND(100,-1,1)
510 CALL COLOR(1,16,14)
520 SC=SC+1
530 CALL HCHAR(G,SPAL,130)
540 GOTO 380
550 REM **HINDERNISSE SETZEN
560 Z=3
570 FOR M=1 TO 100
580 FOR C=1 TO PIT
590 X=INT(32*RND)+1
600 IF Z>24 THEN 650
610 CALL HCHAR(Z,X,ZE)
620 NEXT C
630 Z=Z+T
640 NEXT M
650 RETURN
660 CALL SOUND(2500,131,3,17
5,3,220,3)
670 NEXT T
680 CALL CLEAR
690 FE=FE+1
700 PRINT
710 REM **ASCII-CODE
720 PRINT
730 ZE=ZE+1
740 PRINT
750 PRINT
760 REM HINDERNISZUWACHS
770 PIT=PIT+4
780 REM AUSGABE
790 PRINT "FEHLERPUNKTE :";S
C
800 JU=JU+1
810 IF JU=3 THEN 1020
820 REM FREISPIEL SETZEN!

```

```

830 IF SC<=8 THEN 840 ELSE 9
30
840 PRINT
850 IF SC=0 THEN 860 ELSE 87
0
860 PRINT "OPTIMUM!"
870 PRINT
880 PRINT "***FREISPIEL***"
890 REM PAUSE
900 FOR V=1 TO 800
910 NEXT V
920 GOTO 180
930 PRINT
940 PRINT "C BY HOMECOMPUTER
"
950 PRINT
960 PRINT "NOCH EIN SPIEL? (
J/N)"
970 CALL KEY(0,KEY,STATUS)
980 IF STATUS=0 THEN 970
990 IF KEY=74 THEN 120
1000 STOP
1010 REM ZEICHENSETZUNG AB
RUNDE 8
1020 BU=2
1030 ZE=65
1040 GOTO 830
1050 CALL CLEAR
1060 TR=TR+50
1070 FOR OP=1 TO TR
1080 WA=INT(120*RND)+1
1090 WS=INT(22*RND)+3
1100 WD=INT(32*RND)+1
1110 CALL HCHAR(WS,WD,WA)
1120 NEXT OP
1130 G=0
1140 GOTO 260
1150 END

```



Killersatellit

Sobald das Spiel gestartet wurde, fragt der Computer nach der Wahl des Schiffes. Es stehen 4 Schiffe zur

Wahl, die sich jedoch nur äußerlich unterscheiden. Die Steuerung des Schiffes erfolgt über die 4 Cursor-

Tasten (E, S, D und X). Über die Space-Taste können Schüsse abgegeben werden. Nach jedem erfolgreichem Abschluß erhöht sich die Schwierigkeit und Punktzahl der einzelnen Gegner. Außerdem wird nach je-

dem Abschluß die eigene, sowie die Höchstpunktzahl angezeigt.

Das Spiel ist beendet, wenn der Satellit das Schiff abgeschossen hat. Alles Andere ergibt sich beim Spielen.

Ch. Bladoschewski

```

10 REM *****
20 REM * KILLERSATELLIT *
30 REM * ENTWICKELT VON *
40 REM * CH. BLADOSCHESKI *
50 REM *****
100 CALL CLEAR
110 CALL COLOR(1,2,2)
120 CALL SCREEN(2)
130 PRINT TAB(7);"KILLERSATE
LLIT": " | | | | | | | | | |
140 CALL CHAR(96,"00183C7E7E
E3C18")
150 CALL CHAR(104,"00183C7E7
E3C18")
160 CALL HCHAR(10,6,104,21)
170 CALL HCHAR(14,6,104,21)
180 CALL VCHAR(10,6,104,5)
190 CALL VCHAR(10,26,104,5)
200 FOR I=6 TO 26 STEP 2
210 CALL HCHAR(10,I,96)
220 CALL HCHAR(14,I,96)
230 NEXT I
240 CALL HCHAR(12,6,96)
250 CALL HCHAR(12,26,96)
260 CALL COLOR(1,16,1)
270 CALL SCREEN(16)
280 FOR J=1 TO 20
290 FOR J=1 TO 2
300 CALL COLOR(9,11,9)
310 CALL COLOR(10,5,13)
320 NEXT J
330 CALL COLOR(9,5,9)
340 CALL COLOR(10,11,13)
350 NEXT J
360 CALL COLOR(1,1,3)
370 CALL SCREEN(3)
380 REM
390 SC=0
400 CALL CLEAR
410 RANDOMIZE
420 M=2
430 P1=1
440 U=.1
450 A$="101010388ABAFEB2"
460 R$="42425A665A4242"
470 C$="183C7E3C3C7EFF"
480 D$="82828200828282"
490 E$="1818181818181818"
500 B$="1818183C66FFFF66"
510 G$="42425A5A7EFF42"
520 INPUT "WELCHES SCHIFF WI
LLST DU (1-4)?":SCHIFF
530 IF (SCHIFF<1)+(SCHIFF>4)
<0 THEN 520
540 ON SCHIFF GOTO 640,580,6
10,550
550 Z$=R$
560 Y$=E$
570 GOTO 670
580 Z$=B$
590 Y$=E$
600 GOTO 670
610 Y$=D$

```

```

620 Z$=G$
630 GOTO 660
640 REM
650 Z$=A$
660 Y$=D$
670 X=10
680 Y=10
690 CALL CLEAR
700 B=12
710 A=16
720 CALL SCREEN(2)
730 CALL COLOR(1,1,1)
740 CALL COLOR(9,4,1)
750 CALL COLOR(10,14,1)
760 CALL COLOR(11,9,1)
770 CALL COLOR(12,5,1)
780 CALL CHAR(96,Y$)
790 CALL CHAR(104,"5A5A5A5A5
A5A5A5A")
800 CALL CHAR(112,C$)
810 CALL CHAR(120,Z$)
820 X=X-(INT((RND*2)-.5))
830 Y=Y-(INT((RND*2)-.5))
840 IF X<1 THEN 1240
850 IF X>24 THEN 1260
860 IF Y<1 THEN 1280
870 IF Y>32 THEN 1300
880 CALL KEY(D,K,S)
890 IF K=32 THEN 1320
900 IF K=83 THEN 1040
910 IF K=68 THEN 1070
920 IF K=88 THEN 1100
930 IF K=69 THEN 1130
80
940 IF B<1 THEN 1160
950 IF B>24 THEN 1180
960 IF A<1 THEN 1200
970 IF A>32 THEN 1220
980 CALL CLEAR
990 CALL HCHAR(B,A,120)
1000 CALL HCHAR(X,Y,112)
1010 IF X>21 THEN 820
1020 IF RND<U THEN 1580
1030 GOTO 820
1040 A=A-1
1050 CALL SOUND(-500,-6,0)
1060 GOTO 940
1070 A=A+1
1080 CALL SOUND(-500,-6,0)
1090 GOTO 940
1100 B=B+1
1110 CALL SOUND(-500,-6,0)
1120 GOTO 940
1130 B=B-1
1140 CALL SOUND(-500,-6,0)
1150 GOTO 940
1160 B=B+24
1170 GOTO 980
1180 B=B-24
1190 GOTO 980
1200 A=A+32
1210 GOTO 980
1220 A=A-32

```

```

1230 GOTO 980
1240 X=X+24
1250 GOTO 880
1260 X=X-24
1270 GOTO 880
1280 Y=Y+32
1290 GOTO 880
1300 Y=Y-32
1310 GOTO 880
1320 FOR P=B-1 TO B-M STEP -
1330 CALL SOUND(100,-5,0)
1340 CALL GCHAR(P,A,L)
1350 IF L=112 THEN 1390
1360 CALL HCHAR(P,A,96)
1370 NEXT P
1380 GOTO 980
1390 CALL CLEAR
1400 PRINT "GEGNER ZERSTOERT
| | | | |
1410 SC=SC+P1
1420 IF HS>=SC THEN 1490
1430 HS=SC
1440 FOR SJ=1 TO 5
1450 FOR SJ=-3 TO -1
1460 CALL SOUND(100,SJ,0)
1470 NEXT SJ
1480 NEXT SJ
1490 PRINT "HIGHSCORE:";HS
1500 PRINT "PUNKTZAHL:";SC:
| | | | |
1510 CALL COLOR(1,3,3)
1520 CALL SCREEN(3)
1530 P1=P1+1
1540 U=U+.02
1550 FOR I=1 TO 1000
1560 NEXT I
1570 GOTO 540
1580 FOR P=X+1 TO X+3
1590 CALL SOUND(100,-3,2)
1600 CALL GCHAR(P,Y,L)
1610 IF L=120 THEN 1650
1620 CALL HCHAR(P,Y,104)
1630 NEXT P
1640 GOTO 820
1650 CALL CLEAR
1660 PRINT TAB(5);"DU WURDES
T ZERSTOERT":| | | | |
"GAME OVER":| | | | |
1670 PRINT "HIGHSCORE:";HS
1680 PRINT "PUNKTZAHL:";SC:
| | | | |
1690 CALL COLOR(1,3,3)
1700 CALL SCREEN(3)
1710 PRINT "WILLST DU NOCH E
INMAL SPIE- LEN? DRUECKE (J)
(N)"
1720 CALL KEY(0,K,S)
1730 IF S=0 THEN 1720
1740 IF K=ASC("J") THEN 380
1750 IF K<>ASC("N") THEN 1710
1760 END

```


Eidechse Sharp MZ-80K

Das Programm EIDECHSE wurde auf einem Sharp MZ-80 K geschrieben und ist ungefähr 5,5 KB lang. Die Grundidee ist einem Buch entnommen und wurde von mir erweitert und an den MZ-80 K angepaßt.

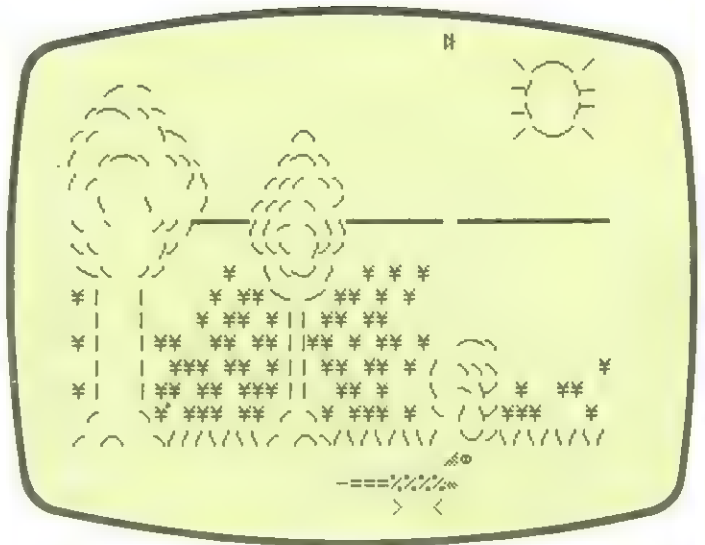
Zum Verlauf des Spieles sei gesagt, es ist ein sehr einfaches Spiel.

Der Spieler soll versuchen, mittels einer beliebigen

Taste die Fliege zu fangen. Auf Tastendruck schnellt die Zunge der Eidechse nach oben und mit etwas Glück oder Geschick trifft er die Fliege.

Nach einer gewissen Trefferquote gibt es einen Zeitbonus. Die Zeit, Treffer und Durchgänge werden dem Spieler auf dem Bildschirm angezeigt.

Egon Kwopil



Die Eidechse mit der Fliege links von der Sonne

```
10 PRINT"EGON KWAPIL HAMBURG 03.1983"
```

```
20 MUSIC"R7R7R7"
```

```
30 PRINT"@"
```

```
40 PRINT"
```

```
50 PRINT"
```

```
60 PRINT"
```

```
70 PRINT"
```

```
80 PRINT"
```

```
90 PRINT"
```

```
100 PRINT"
```

```
110 PRINT"
```

```
120 USR(68):NEXTA:PRINT"
```

```
130 PRINT"
```

```
140 PRINT"
```

```
150 PRINT"
```

```
160 PRINT"
```

```
170 PRINT"
```

```
180 PRINT"
```

```
190 PRINT"
```

```
200 PRINT"
```

```
210 PRINT"
```

```
220 PRINT"
```

```
230 PRINT"
```

```
240 PRINT"
```

```
250 PRINT"
```

```
260 A$=" _CR3_DR3_ER3_DR3_DR3_CR3_DR3_ER3_CR3_DR3_ER3_ER3_DR3_DR3_C5"
```

```
270 B$=" _CR3_DR3_ER3_DR3_DR3_CR3_DR3_ER3_CR3_DR3_ER3_ER3_DR3_DR3_C5"
```

```
280 C$=" _CR3_DR3_ER3_DR3_DR3_CR3_DR3_ER3_CR3_DR3_ER3_ER3_DR3_DR3_C5"
```

```
290 TEMPO7:MUSICA$,B$,A$,C$
```

```
300 PRINT"@" EIDECHSE"
```

```
310 PRINT" ====="
```




```

320 REM REAKTIONSSPIEL
330 PRINT
340 PRINT "Die Erdechse versucht, die Fliese zu"
350 PRINT "Erhaschen. Zum Betätigen der Taste Z"
360 PRINT "drücken Sie irgendetwas Taste."
370 PRINT "Sie haben dazu 3 Minuten Zeit."
380 PRINT "Bei 10, 20 oder 30 Punkten gibt es
390 PRINT "Überraschung. Die Fliese sieht so aus und"
400 PRINT ".....
410 PRINT "Ist schwer zu finden !!!
420 PRINT ".....
430 PRINT "..... U I E L S P I E L
440 PRINT "Drücken Sie jetzt eine Taste."
450 SEIT$=IET$:"THEM$=50
460 REM VORBEREITUNG
470 PRINT "G":MUSIC$
480 PRINT TAB(13);"XXXXXXXXXXXXXXXXXXXX"
490 REM
500 REM UORWARZEIT
510 REM
520 FOR I=1 TO 1000: NEXT I
530 REM
540 REM UHR WIRD GESTELLT
550 REM
560 TI$="000000"
570 REM
580 REM FLIEGENZAHLER
590 REM
600 Z=0:F=0
610 FOR A=1 TO 200: POKE 4514, A: USP(68)=NEXT
620 REM
630 REM SPIEL
640 REM
650 PRINT "G":F=F+1:REM RUNDENZAHLER
660 POKE 4466, 5
670 PRINT "
680 PRINT "
690 PRINT "
700 PRINT "
710 PRINT "
720 PRINT "
730 PRINT "
740 PRINT "
750 PRINT "
760 PRINT "
770 PRINT "
780 PRINT "
790 PRINT "
800 PRINT "
810 PRINT "
820 PRINT "
830 PRINT "
840 REM
850 REM FLIEGE PLAZIEREN
860 REM
870 R=(10+RND(1))
880 X=5328+11*R+20+(25*RND(1))
890 POKE X, 231
900 REM
910 REM EIDECHSE LAUFEN LASSEN
920 REM
930 FOR K=54168 TO 54206: REM KOPF VON EID
940 POKE K-1, 0: POKE K, 0: POKE K-41, 0: POKE K-40, 0
950 POKE K, 212: POKE K-1, 101: POKE K-2, 101: POKE K-3, 101
960 POKE K-4, 101: POKE K-5, 43: POKE K-6, 43: POKE K-7, 43
970 POKE K-8, 42: POKE K-9, 216: POKE K-10, 131: POKE K-11, 0
980 POKE K+39, 81: POKE K+38, 0: POKE K+36, 87: POKE K+35, 0
990 REM
1000 REM ZEITANSAGE
1010 REM
1020 POKE 4466, 1: POKE 4465, 15: PRINT "FLIEGEN :":Z: "
1030 POKE 4466, 1: PRINT "ZEIT":":TI$:"
1040 TI=URL+TI$
1050 IFTI>000300 THEN I=50

```



Chiffrier/ Dechiffrier- Programm

TI 99/4A + Drucker

Eingabe

Das Pgm fragt zunächst, ob dechiffriert werden soll, - dann nur DE eingeben (wird im Pgm auch gesagt) Ansonsten verlangt es die Eingabe einer vierstelligen Codezahl (z.B. Datum in der Form 3103 = 31.03.) Diese Codezahl muß auch dem Entschlüssler bekannt sein.

Das Pgm nimmt nur Zahlen an und zwar nur eine! Es sind also alle vier ein-

zeln einzugeben. Danach wird die Eingabe einer zu verschlüsselnden Zeile verlangt. Das Pgm nimmt nur 28 Zeichen an und sagt an, daß der Zeilenrest ggfls mit Punkten zu füllen ist. Die Eingabe geschieht ganz normal und der Drucker druckt sie verschlüsselt aus. So kann Zeile für Zeile eingegeben werden. Die letzte Eingabe soll eine 0 sein, damit das Pgm erkennt, daß die Ein-

gabe abgeschlossen ist. Das Dechiffrieren geschieht genauso, nur daß hier eben der chiffrierte Text eingegeben wird, der dann Zeile für Zeile dechiffriert herauskommt. Interessant und daher kaum zu „knacken“ ist der Algorithmus des Pgms hinsichtlich der Verschlüsselung. Manchmal ist der gleiche Buchstabe von gleicher Bedeutung, manchmal haben 2 gleiche Buchstaben eine völlig andere Bedeutung und manchmal werden für einen Buchstaben im Laufe eines längeren Textes bis zu 50 andere eingesetzt.

Programmbeispiel:

Eingegeben wurde zeilenweise

DIES IST EIN CHIFFRIERTEST:
AUFGENOMMEN AM

31.03.83 FÜR DIE ZEITSCHRIFT HOMECOMPUTER

Da die Eingabe automatisch auf 28 Zeichen beschränkt ist, können Sie am Zeilenende jeweils auf dem . stehenbleiben, der Computer meldet sich mit einem BEEP.

So sah der Ausdruck aus:

```
-----  
DIES IST EIN CHIFFRIERTEST:  
AUFGENOMMEN AM  
-----
```

und als der vorige Ausdruck Zeichen für Zeichen, wiederum zeilenweise nach vorheriger Eingabe der Codeziffern eingegeben wurde, kam folgender Ausdruck zustande:

```
-----  
DIES IST EIN CHIFFRIERTEST:  
AUFGENOMMEN AM  
-----
```

```
100 CALL CLEAR :: CALL SCREEN(16  
)  
110 DIM B$(30)  
120 DIM B(30)  
130 INPUT "ZUM DECHIFFRIEREN DE  
EINGEBEN":DE$ :: IF DE$="DE" THE  
N 370  
140 PRINT "GIB JETZT EINZELN DIE  
4 CODEZIFFERN EIN:"  
150 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE 1:W  
160 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE 1:W  
170 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE 1:W  
180 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE 1:W  
190 PRINT "NENTZEILE EINGEBEN (3  
200 ACCEPT BEEP SIZE(28):A$  
210 IF A$="0" THEN END  
220 FOR I=1 TO 28  
230 B(I)=ASC(SEG$(A$,I,1))  
240 NEXT I  
250 FOR I=1 TO 28 STEP 4 :: B(I)  
=B(I)+W :: NEXT I  
260 FOR I=2 TO 28 STEP 4 :: B(I)  
=B(I)+X :: NEXT I  
270 FOR I=3 TO 28 STEP 4 :: B(I)  
=B(I)+Y :: NEXT I  
280 FOR I=4 TO 28 STEP 4 :: B(I)
```

```
=B(I)+Z :: NEXT I  
290 FOR I=1 TO 28  
300 IF B(I)≠32 THEN B(I)=32  
310 NEXT I  
320 FOR I=1 TO 28 :: B$(I)=CHR$(  
B(I)):: NEXT I  
330 OPEN #1:"TP.E":OUTPUT  
340 FOR I=1 TO 28 :: PRINT #1:B$(  
I):: NEXT I  
350 CLOSE #1  
360 GOTO 200  
370 PRINT "GIB JETZT EINZELN DIE  
4 CODEZIFFERN EIN:"  
380 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE(1):W  
390 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE(1):X  
400 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE(1):Y  
410 ACCEPT BEEP VALIDATE(DIGIT)$  
IZE(1):Z  
420 PRINT "GIB DEN ZEILENTEXT EI  
N":A$  
430 ACCEPT BEEP SIZE(28):A$  
440 IF A$="0" THEN END  
450 FOR I=1 TO 28  
460 B(I)=ASC(SEG$(A$,I,1))  
470 NEXT I  
480 FOR I=1 TO 28 STEP 4 :: B(I)  
=B(I)+W :: NEXT I  
490 FOR I=2 TO 28 STEP 4 :: B(I)  
=B(I)+X :: NEXT I
```



```

500 FOR I=3 TO 28 STEP 4 :: B(I)
=B(I)+Y :: NEXT I
510 FOR I=4 TO 28 STEP 4 :: B(I)
=B(I)+Z :: NEXT I
520 FOR I=1 TO 28
530 IF B(I)<=41 THEN B(I)=32
540 NEXT I
550 FOR I=1 TO 28 :: B$(I)=CHR$(

```

```

B(I)):: NEXT I
560 OPEN #1:"TP.E",OUTPUT
570 FOR I=1 TO 28 :: PRINT #1:B$(
(I):: NEXT I
580 CLOSE #1
590 GOTO 430
600 END

```

Rangliste

TI 99/4A + Drucker

Aufgrund einer Wahl oder eines Wettbewerbes soll im Verein z. B. eine Rangliste der Ergebnisse aufgestellt werden.

Die Verarbeitungszeit für 100 Teilnehmer beträgt ca. 5 min.

Das Programm fragt in ei-

ner Dreiergruppe für jeden Teilnehmer:

LISTENPLATZ
NAME

ERGEBNIS (in Punkten oder Stimmen)

Sind die Angaben für alle Teilnehmer eingegeben, erwartet das Pgm zum Abschluß unter LISTENPLATZ die Eingabe einer 0

und beginnt die Verarbeitung.

Ausgabeform:

1. (5.) Mayerhofer = 189
dabei ist die erste Zahl der Ranglistenplatz die in () stehende der bisherige Listenplatz und nach dem Namen folgt das Ergebnis aus den Punkten/Stimmen. Das Pgm stoppt - zur Verlesung der Ergebnisse - automatisch nach der Ausgabe von jeweils 5 Ergebnissen und kann dann durch Drücken irgendeiner Taste

fortgesetzt werden.

Es erfolgt die Ausgabe eines kompletten Druckprotokolls.

Falls mehr als 99 Teilnehmer erfaßt werden sollen, sind die DIM-Statements in den Zeilen 110-120-130 entsprechend der Teilnehmerzahl + 1 zu erhöhen.

Das Programm läuft auf dem TI 99/4 A mit Extended-Basic-Modul + Drucker.

Herbert May

```

1 CALL SCREEN(16)
2 DIM LP(100)
100 DIM CY$(100)
110 DIM A(100)
120 FOR I=1 TO 100
121 PRINT
122 INPUT "LISTENPLATZ   ":LP(I)
123 IF LP(I)=0 THEN 220
124 INPUT "NAME           ":CY$(
I)
125 INPUT "ERGEBNIS     ":A(I)
126 N=N+1
127 LP(I)=LP(I)+100
128 A(I)=A(I)+LP(I)
129 NEXT I
220 GOSUB 320
221 CALL CLEAR
230 FOR I=1 TO 100
240 AA=SEC A(I)-INT(A(I)/
250 A(I)=INT(A(I)/
260 AA=AA*100
270 B=B+1
275 PRINT :
280 PRINT "B$(I)";A(I);TAB(12)
;CY$(I);":":TAB(25);A(I)
285 OPEN #1:"TP.E",OUTPUT :: PRI
NT #1:B$;";AA;":":TAB(12);CY$(
AA;":":TAB(25);A(I):: CLOSE #1
290 Z=Z+1
300 IF Z=5 THEN 590
301 IF AA=0 THEN 311
310 NEXT I
311 END
320 HL=N

```

```

330 FOR HK=INT(N/2) TO 1 STEP -1
340 HI=HK
350 GOSUB 460
360 NEXT HK
370 FOR HK=N TO 2 STEP -1
380 HH=A(I)
390 A(I)=A(HK)
400 A(HK)=HH
410 HI=1
420 HL=HK-1
430 GOSUB 460
440 NEXT HK
450 RETURN
460 HJ=2+HI
470 IF HJ>HL THEN 570
480 IF HJ=HL THEN 510
490 IF A(HJ)<A(HJ+1) THEN 510
500 HJ=HJ+1
510 IF A(HJ)>A(HI) THEN 570
520 HH=A(HI)
530 A(HI)=A(HJ)
540 A(HJ)=HH
550 HI=HJ
560 GOTO 460
570 RETURN
580 Z=Z+1
590 Z=0
600 GOSUB 620
610 GOTO 310
620 CALL KEY(J,ST)
630 IF ST=0 THEN 620
640 IF J=74 THEN 650
650 RETURN
660 Z=0

```


Starship VC-20



HC-Invaders

Zum ersten populären Computer-Videospiel wurden weltweit die Space-Invaders, kleine Monster aus dem All, die versuchen, die Erde zu erobern. Sie haben es geschafft; keine Frage!

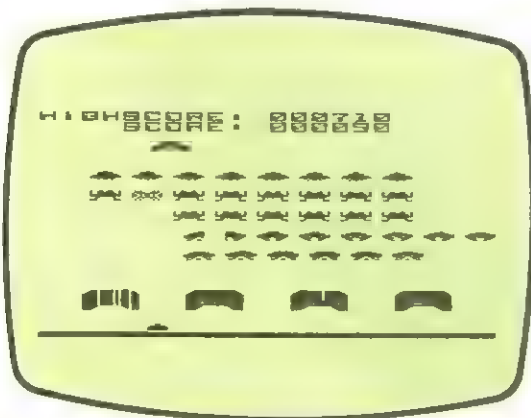
Homecomputer bringt Ihnen für Ihren VC-20, damit Sie Gelegenheit erhalten, den Editor/Assembler zu testen, eine eigene Version der bekannten Invaders ins Haus.

Instruktionen:

1. Editor eingeladen
2. INVADERS (Quellprogramm) eingeben
3. Assembler einladen
4. POKE 55,254: POKE 56,55: RUN
5. POKE 43,254: POKE 44,55: POKE 45,64: POKE 46,63: POKE 56,64: POKE 55,0
6. SAVE „INVADERS“ (MASCHINENPROGRAMM)
7. POKE 43,1: POKE 44,28: POKE 168,0:
8. LOAD „ZEICHENDATAS“
9. RUN
10. LOAD „INVADERS“ (BASIC-PGM)
11. RUN

Wenn alles auf Kassette gespeichert ist, geht man vor wie folgt: POKE 43,254: POKE 44,55: „LOAD“ (Maschinenprogramm).

LOAD „INVADERS“ (Maschinenprogramm) fahre fort wie oben ab 7.



ZEICHENDATAS

```

10 FORI=5120T06511 REDE POKEI, E NEXT
1000 DATA124,64,124,68,124,12,12,12,66,126,98,98,98,98,0
1010 DATA124,36,62,50,50,50,126,0,126,66,64,96,96,98,126,0
1020 DATA126,34,50,58,50,126,0,126,84,64,120,36,96,126,0
1030 DATA126,64,64,120,96,96,96,0,126,66,64,110,98,98,126,0
1040 DATA66,66,66,126,98,98,98,0,161,16,24,24,24,24,0
1050 DATA2,2,6,5,70,60,0,66,68,72,126,98,98,98,0
1060 DATA64,64,64,96,96,126,0,102,90,66,98,98,98,98,0
1070 DATA11,74,74,106,106,102,0,126,66,66,98,98,98,98,0
1080 DATA126,66,66,126,96,96,96,0,126,66,66,96,96,98,126,0
1090 DATA126,66,66,126,100,98,98,0,126,66,64,126,6,70,126,0
1100 DATA124,16,16,24,24,24,24,0,66,66,66,98,98,98,126,0
1110 DATA66,66,66,36,24,24,24,0,98,98,98,98,66,98,102,0
1120 DATA66,36,24,124,98,98,98,0,66,66,66,60,24,24,0
1130 DATA126,2,28,96,96,126,0,60,32,32,48,48,48,60,0
1140 DATA60,126,255,255,255,255,126,60,60,4,12,12,12,60,0
1150 DATA0,8,28,42,16,8,0,0,16,32,127,32,16,0
1160 DATA0,0,0,0,0,0,24,24,28,28,28,0,28,0
1170 DATA108,108,36,36,0,0,36,56,126,52,126,52,52,0
1180 DATA80,82,40,62,26,62,24,0,112,82,116,8,30,42,78,0
1190 DATA120,72,48,112,106,108,122,0,56,56,8,0,0,0,0
1200 DATA12,16,32,48,48,16,12,0,48,0,4,12,12,8,48,0
1210 DATA0,42,28,62,42,0,0,0,62,62,12,12,0
1220 DATA0,0,0,0,56,56,8,0,0,126,126,0,0,0
1230 DATA0,0,0,0,24,24,0,0,2,4,8,16,32,64,0
1240 DATA126,66,78,106,114,98,126,0,56,8,6,24,24,24,60,0
1250 DATA126,66,72,126,96,96,126,0,62,2,2,36,6,6,126,0
1260 DATA64,64,68,68,126,12,0,62,32,62,6,70,126,0
1270 DATA126,64,64,126,98,98,126,0,126,66,4,8,24,24,24,0
1280 DATA60,36,36,24,98,98,126,0,126,66,65,126,6,6,126,0
1290 DATA0,0,24,0,24,0,0,0,56,0,56,0,56,0,0,0
1300 DATA14,24,48,56,48,4,4,0,0,126,0,126,126,0,0
1310 DATA11,24,12,6,12,24,112,0,126,66,2,62,48,0,48,0
1320 DATA6,31,31,102,255,47,198,0,128,128,96,240,64,48,0
1330 DATA0,1,6,15,1,0,96,248,248,102,255,244,104,0
1340 DATA25,63,47,255,191,160,153,0,133,133,64,240,208,80,144,0
1350 DATA9,9,10,15,8,2,14,0,153,245,245,252,4,7,0
1360 DATA15,63,255,230,121,16,182,0,192,240,112,24,128,96,0
1370 DATA0,8,15,14,7,3,1,0,240,252,255,103,158,12,152,0
1380 DATA0,0,24,24,126,255,255,255,0,0,0,0,0,0,0
1390 DATA0,0,6,31,53,63,63,0,0,0,128,192,192,192
1400 DATA0,0,1,7,5,15,15,0,124,124,224,240,240,240
1410 DATA0,0,0,0,1,9,9,3,0,96,96,240,252,252,252
1420 DATA3,15,31,63,106,255,112,32,192,240,248,252,86,255,14,4
1430 DATA0,0,0,0,0,0,0,1,7,15,31,53,127,56,16
1440 DATA224,248,252,254,4,255,7,2,0,0,0,0,0,0,0
1450 DATA0,3,7,15,26,63,28,8,240,252,254,255,149,255,3,1
1460 DATA0,0,0,0,128,192,128,0,0,1,3,7,13,31,14,4
1470 DATA120,254,255,255,74,255,1,0,0,0,8,128,192,224,192,128
1480 DATA0,0,1,3,6,15,7,2,60,255,255,255,165,255,0,0
1490 DATA0,0,128,192,96,240,224,64,0,0,1,3,7,3,1
1500 DATA30,127,255,255,80,255,126,0,0,128,192,224,176,248,112,32
1510 DATA0,0,0,1,3,1,0,15,63,127,255,163,255,192,128
1520 DATA0,192,224,248,88,252,56,16,0,0,0,0,1,0,0
1530 DATA7,31,63,127,225,224,64,128,224,248,248,172,254,28,8
1540 DATA0,41,50,64,150,41,80,0,160,64,144,32,144,64,160,0
1550 DATA5,2,9,4,2,0,0,10,148,105,2,105,148,10,0
1560 DATA124,234,127,181,33,0,0,0,0,0,0,0,0,0,0,0
1570 DATA31,63,31,45,8,0,0,0,128,192,64,64,0,0,0
1580 DATA7,15,7,11,2,0,0,0,192,224,248,80,16,0,0,0
1590 DATA1,3,1,2,0,0,0,0,240,243,252,212,192,0,0,0
1600 DATA164,85,171,106,60,126,126,60,0,0,0,0,0,0,0
1610 DATA10,5,10,5,3,7,3,64,80,176,160,192,224,224,192
1620 DATA144,94,85,40,44,58,127,255,0,0,0,0,0,0,0
1630 DATA35,8,21,10,11,14,31,63,8,128,64,0,0,192,192
1640 DATA9,5,2,2,3,7,15,0,32,98,128,192,160,240,240
1650 DATA2,0,1,0,0,0,1,3,64,156,84,160,176,232,252,252
1660 DATA9,68,170,20,52,126,254,255,0,0,0,0,0,0,0
1670 DATA2,17,42,5,13,23,63,66,64,0,128,0,0,0,128,192
1680 DATA0,4,10,1,3,5,15,15,144,64,160,64,64,152,224,248
1690 DATA0,1,2,0,0,1,3,36,16,168,80,168,112,248,252
1700 DATA15,31,63,127,127,127,255,255,255,255,255,255,255
1710 DATA240,248,252,254,254,254,254,15,31,63,127,127,127
1720 DATA255,255,255,255,255,255,255,240,248,252,254,254,254
  
```

READY.

INVADERS (BASIC-PGM)

```

50 POKES5,254 POKES5,57
100 FORF=130P,0 POKES101,0 FORI=5122T05127 POKEI,255 NEXT REM BODEN IM CHR-RAM
110 POKES101,0 REM NEUER BOIEN
120 FORI=752,2 REM HIMMELRLINGGESCHWINDIGKEIT
130 POKES806,3 REM BODENBLINKGESCHWINDIGKEIT
140 POKE14812,15 FEMUEMELINKRANZLH
150 POKE14316,4 REM HIMMELRLINKRANZLH
160 POKE16190,201 POKE16191,64 POKE16192,144 POKE16193,5 POKE16194,201 POKE16195
  76
170 POKE16196,176 POKE16197,1 POKE16198,56 POKE16199,104 POKE16200,24 POKE16201,
  105
180 POKE16202,4 POKE16203,72 POKE16204,94
190 POKE16205,204 POKE16206,32 POKE16207,62 POKE16208,63
200 POKE16209,234 POKE16210,32 POKE16211,62 POKE16212,63
210 POKE16213,2 REM INVAIERGESCHWINDIGKEIT
220 POKE16018,66 REM BLAUE FARBE
230 POKE15931,3 REM INVAIERLINKRANZLH
240 GOTO270 SCHRUEKARR,UP
250 S=0 FORI=4130T01-57,TEF-1 S=S0WS+(15RNDPEEK(I)) NEXT
260 IF$HIMMELRLING
270 FORI=37868+66T01+21 POKEI,5 NEXT
280 FORI=37354T01+21 POKEI,53 NEXT
290 FORI=62665,207
300 PRINT"MAIORSCHAU RIGHT$(STR$(I55+HI)) 6)
310 PRINT" STAGE 000000
320 PRINT"#####
330 POKE12,1 REM KRANZLH BRSEN
340 FORI=37955T01+19 POKEI,4 NEXT
350 S1=15309 REM INVADER+MOVE
360 S2=15339 REM UFO
370 S3=15103 REM TASTAT-ABFRAGE
380 S4=14853 REM SPIELEZEICHENS WETTERBEWEGEN
390 S5=15740 REM HIMMELRLINKEN
400 S6=15794 REM ROEMERLINKEN
410 S7=15848 REM INVADERLINKEN
420 S8=14326 REM INVADERSCHUSS INITIALISIEREN UND WETTERBEWEGEN
430 S9=15659 REM BASTOFLINKEN
440 FORI=932T0860 POKEI,0 NEXT
450 FORI=864T1+44 POKEI,256RND(I) NEXT
460 POKE82,80 POKE83,16 POKE89,88 POKE90,16 POKE36869,205
470 POKE81,16 POKE82,4 POKE83,1 POKE84,64
480 SYS15936 POKES,0
490 FORI=97888+21*22T01+21 POKEI,2 NEXT
500 IFPEEK(3) THEN250
510 IFPEEK(861)>A THENGOSUB9550
520 RI=PEEK(861)
530 SYSS1=PEEK(862) SYSS3=SYSS4 SYSS5=SYSS6 SYSS7=SYSS8 SYSS9
540 GOTO500
550 RESTORE FORI=624T01+23 READR POKEI,R POKEI+24,R POKEI+48,R POKEI+72,R NEXT
560 FORI=6416T01+23 READR POKEI,R POKEI+24,R POKEI+48,R POKEI+72,R NEXT
570 PRINT"#####
580 PRINT"CHR$(3R) POKE212,0 PRINT"CHR$(8R) CHR$(8R) CHR$(8R)
590 RETURN
600 DATA15,31,63,127,127,127,127,255,255,255,255,255,255,255,255,255,255
610 DATA240,248,252,254,254,254,254,15,31,63,127,127,127,127,127,127,127
620 DATA27,127,127,127,127,127,127,126,124,120,255,255,255,255,0,0,0,0
630 DATA254,254,254,254,254,254,254,254,254,254,254,126,62,30
  
```


Starship VC-20

INVADERS (Quellprogramm)

```

1      ORG $37FE
2PTR1 EQU $33C
3PTR2 EQU PTR1+1
4PTR3 EQU PTR2+1
5PTR4 EQU PTR3+1
6UFF  EQU $340
7UP   EQU $341
8BOBZ EQU $342
9BBZ  EQU $343
10ISF EQU $344
11ZUISCH EQU $345
12ZUSSCH EQU $346
13INVI EQU $347
14ERG  EQU $351
15BP   EQU $352
16TAB  EQU $353
17SSF  EQU $357
18ISCHADR EQU 0
19SSCHADR EQU 193
20HIM  EQU $356
21HIME EQU $357
22SBDZ EQU $354
23BJ   EQU $356
24IKC  EQU $35C
25AZI  EQU $35D
26INVA EQU $35E
27RZB  EQU $390
28LIZ  EQU $391
29MR   EQU $34C
30R0Z  EQU $341
31M0C  EQU $34E
32S0C  EQU $34E
33M0A  EQU $03E+PTR1
34BL0  EQU $03E
35P1   EQU 1
36P2   EQU 2
37P3   EQU 3
38     DEW
39     DEW
40PHD  LDX PTR1
41     LDX $360, X
42     LDX PTR2, X
43     LDX $360, X
44     LDX PTR3, X
45     ADC $37C, X
46     LDX PTR4, X
47     STA $360, X
48     PHX
49     LDY #0
50     INC PTR1
51     LDA PTR1
52     CMP #17
53     BNE *+3
54     STY PTR1
55     INC PTR2
56     LDA PTR2
57     CMP #15
58     BNE *+3
59     STY PTR2
60     INC PTR3
61     LDA PTR3
62
63
64
65
66
67
68
69
70
71
72CHRRADR
73
74
75
76
77
78
79
80
81
82
83
84
85
86SCO
87
88
89
90
91SC1
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121SC2
122

```

```

CMP #17
BNE *+3
STY PTR3
INC PTR4
LDA PTR4
CMP #45
BNE *+3
STY PTR4
PLA
RTS
LDY #0
STY $FF
ASL
ROL $FF
ASL
ROL $FF
ASL
ROL $FF
STA $FE
CLC
LDA $FF
ADC #20
STA $FF
RTS
STA $7
SED
CLC
LDX #4
LDY #0
ADC $7
BCD *+2
INY
CLC
DEX
BNE SC1
STA $7
CLD
AND #15
ADC SC+6
STA SC+6
CMP #$3A
BCD *+5
SBC #10
STA SC+6
PHP
LDA $7
LSR
LSR
LSR
PLP
ADC SC+5
STA SC+5
CMP #$3A
BCD *+5
SBC #10
STA SC+5
TYA
LDX #4
ADC #0
ADC SC, X

```


Starship VC-20

123	STA SC,X	184Z57	STA 6528,X
124	CMP #3A	185	DEX
125	BCC **5	186	BPL Z57
126	SBC #10	187	STX ISF
127	STA SC,X	188	TYR
128	LDA #0	189	CLC
129	DEX	190	ADC #22
130	BNE SC2	191	ADC ISCHADR
131	RTS	192	STA ISCHADR
132INWADER	LDA BOBZ	193	BCC **2
133	ORA BBZ	194	INC ISCHADR+1
134	BEQ **1	195	LDA #32
135RTS	RTS	196	STA ZUISCH
136	LDA ISF	197	LDY #0
137	BNE ISCH	198	LDA #176
138	JSR PND	199	STA (ISCHADR),Y
139	CMP #8	200	RTS
140	BCS RTS	201 ISCH	LDY #0
141	LDX INVR+1	202	LDA ZUISCH
142	LDA INVR	203	STA (ISCHADR),Y
143	CLC	204	CLC
144	ADC #176	205	LDA ISCHADR
145	STA #FE	206	ADC #22
146	ROL **1	207	STA ISCHADR
147	INX	208	BCD **2
148	STX #FF	209	INC ISCHADR+1
149	JSR END	210	LDA (ISCHADR),Y
150	AND #1F	211	STA ZUISCH
151	CMP #0	212	CMP #7
152	BCS +-9	213	BNE JS91
153	TYR	214	LDA #176
154IS1	LDA #FE	215	STA (ISCHADR),Y
155	STA ISCHADR	216	RTS
156	LDA #FF	217JS91	CMP #76
157	STA ISCHADR+1	218	BCC P3
158	LDX #5	219	CMP #140
159IS2	LDA (0),Y	220	BCS **4
160	CMP #64	221	CMP #64
161	BCS IS3	222	BCS P3
162	SEC	223	CMP #174
163	LDA ISCHADR	224	BCS P3
164	SBC #44	225	JSR CHRPRDR
165	STA ISCHADR	226	LDY #7
166	BCS **2	227X20	LDA (#FE),Y
167	DEC ISCHADR+1	228	STA 6536,Y
168	DEX	229	AND 6528
169	BNE IS2	230	BNE X21
170	BEQ JS4	231	DEY
171IS3	AND #3	232	BPL X20
172	CLC	233	LDY #7
173	BEQ IS5	234X22	LDA (#FE),Y
174	CMP #3	235	ORA 6528
175	BEQ IS5	236	STA 6536,Y
176IS4	DEY	237	DEY
177	BPL IS1	238	BPL X22
178	LDY #21	239	INY
179	BNE IS1	240	LDA #177
180IS5	BCC **3	241	STA (ISCHADR),Y
181	LDA #40	242	RTS
182	DFB #20, #A9, 4	243X21	LDY #0 ;GETROFF
183	LDX #7	244	STY ISF

Starship VC-20

245	LDA ZUISCH	305	SBC #22
246	CMR #84	307	STA SSCHADR
247	BCS X23	308	BCS *+2
248	ADC #48	309	DEC SSCHADR+1
249	TAY	310	CMR #66
250	LDA BP	311	LDA SSCHADR+1
251	LSR	312	SBC ##10
252	LSR	313	BCS SF2
253	TAX	314	LDM HIM
254	TYA	315	CPX #3
255	STA BLA,X	316	BCC *+8
256	CLC	317	LDA SSCHADR
257	ADC #1	318	BNE *+2
258	STA BLA+1,X	319	DEC SSCHADR+1
259	LDA #10	320	DEC SSCHADR
260	STA BBC	321	TXA
261	STA BDC	322	ASL
262	RTS	323	CLC
263X23	LDA #52X	324	STY SSF
264	FOR ##FF	325	ADC #1,2
265	LDY #7	326	STA (SSCHADR),Y
266	PHA	327	INY
267X51	PLA	328	ADC #1
268	PHA	329	STA (SSCHADR),Y
269	AND (##FF),Y	330	LDM #20
270	STA (##FF),Y	331	STX HINC
271	DEY	332	STX SMIZ
272	AND #0	333	RTS
273	PLA	334XFD	LDY #4
274	TLY	335	LDA (SSCHADR),Y
275	TYA JSR	336	STA ZUSSCH
276	RTS	337	CMR #32
277X1	LDM #20 ;BODEN	338	BNE *+5
278	STX RORL	339	LDA #178
279	STX BBC	340	STA (SSCHADR),Y
280	LDY #0	341	RTS
281	STY JSR	342	JSR CHRADR
282	LDA #528	343	LDY #7
283	CMR #+4	344XFD	LDA (##FE),Y
284	ROR RORL	345	STA #552,Y
285	LDM (SSCHADR)	346	AND #544
286	BNE *+2	347	BNE SFG
287	DEC (SSCHADR+1)	348	DEY
288	DEC (SSCHADR)	349	BPL SFG
289RUI1	LDA #0	350	LDY #7
290	ROL	351SF4	LDA (##FE),Y
291	ASL	352	ORA #544
292	HDC #120	353	STA #552,Y
293	STA (SSCHADR),Y	354	DEY
294	TAX	355	BPL SF4
295	ADC #1	356	INY
296	STA (SSCHADR),Y	357	LDA #178
297	RTS	358	STA (SSCHADR),Y
298SSCHINC	LDA SSF	359	RTS
299	BNE *+1	360SFG	LDA ZUSSCH ;GETROFFEN
300	RTS	361	LDY #0
301	LDY #0	362	STY SSF
302	LDA ZUSSCH	363	CMR #140
303	STA (SSCHADR),Y	364	BCC SF3
304	SEC	365	CMR #174
305	LDA SSCHADR	366	BCS SF5

Starship VC-20

367	LDA 6544	428	TAX
368	EOR #FF	429	LDA #32
369	LDY #7	430	STA BLA,X
370	PHA	431	STA BLA+1,X
371	PLA	432	LDA 197
372	PHA	433	CMP #30
373	AND (#FE),Y	434	BNE #+10
374	STA (#FE),Y	435	LDY BP
375	DEY	436	CPY #83
376	BPL #-9	437	BEQ #+3
377	PLA	438	INC BP
378	RTS	439	CMP #33
379SF5	CMP #76	440	BNE #+8
380	BCC SF6	441	LDY BP
381	STA BI	442	BEQ #+3
382	LSR	443	DEC BP
383	BCC SF7	444	LDA BP
384	LDA SSCHADR	445	LSR
385	BNE #+2	446	LSR
386	DEC SSCHADR+1	447	TAX
387	DEC SSCHADR	448	LDA BP
388SF7	LDX #10	449	AND #3
389	STX IB2	450	ASL
390	INC SBIZ	451	ADC #76
391	RTS	452	STA BLA,X
392SF6	LDX #21	453	ADC #1
393	LDA #32	454	STA BLA+1,X
394	STA ULA,X	455T1	RTS
395	JAX	4566S	LDA SSF
396	BNE #-6	457	ORA IB2
397	STX SSF	458	ORA HIM2
398	STX UFF	459	BNE T1
399	TSP RND	460	LDA BP
400	CMP #254	461	AND #3
401	BCC #+4	462	TAX
402	LDA ##60	463	STX HIM
403	BNE JSC	464	LDA TAB,X
404	CMP #252	465	LDY #7
405	BCC #+4	466Z34T	STA 6544,Y
406	LDA ##50	467	DEY
407	BNE JSC	468	BPL Z34T
408	CMP #248	469	LDA BP
409	BCC #+4	470	LSR
410	LDA ##40	471	LSR
411	BNE JSC	472	CPX #3
412	CMP #240	473	SBC #21
413	BCC #+4	474	ADC #>BLA
414	LDA ##30	475	STA SSCHADR
415	BNE JSC	476	LDA #<BLA
416	CMP #224	477	SBC #0
417	BCC #+4	478	STA SSCHADR+1
418	LDA ##20	479	STA SSF
419	BNE JSC	480	LDA #32
420	LDA ##15	481	STA ZUSSCH
421JSC	JMP SC0	482	LDY #0
422TAST	LDA 197	483	LDA #178
423	CMP #32	484	STA (SSCHADR),Y
424	BEQ SS	485Z449	RTS
425	LDA BP	486UFO	LDA UFF
426	LSR	487	BNE INCU
427	LSR	488	JSR RND

Starship VC-20

```

489 CMP #254
490 BCC Z449
491 STA UFF
492 LDA #0
493 STA UP
494 LDA #84
495 STA ULA
496 LDA #85
497 STA ULA+1
498 LDA #86
499 STA ULA+2
500 RTS
501 INCU
502 LSR
503 LSR
504 LSR
505 TRX
506 LDA #32
507 STA ULA+2
508 STA ULA+1
509 STA ULA+0
510 INC UP
511 LDA UP
512 CMP #+3
513 BNE #+4
514 STA UFF
515 RTS
516 CMP #+10
517 LSR
518 LSR
519 LSR
520 LSR
521 LDA UP
522 BNE #+
523 BNE #+1
524 BNE #+2
525 BNE #+3
526 BNE #+4
527 BNE #+5
528 BNE #+
529 STA ULA+1
530 BCC #+
531 STA ULA+2
532 RTS
533 INCU
534 BNE #+
535 RTS
536 LDA IBC
537 BNE #+
538 LDA #+
539 STA MDC
540 LDY LIT
541 DEX
542 BPL #+
543 LDA #4
544 STA LJC
545 RHP
546 LDA IN+R+1
547 STA $FF
548 LDA IN+R
549 IM1

```

```

550 DEX
551 BMI IM2
552 ADC #44
553 BCC IM1
554 INC $FF
555 BCS IM1
556 IM2
557 STA $FE
558 LDY #21
559 LDA #0
559 ORA ($FE),Y
560 BEX
561 BPL #+
562 BNE #+
563 CMP #2
564 LDN #+
565 BNE IM3
566 LDA #+
567 BCC #+
568 STA 88
569 LDA $FF
570 BNE #+
571 STA 80
572 IM4
573 STA #+
574 BNE #+
575 BNE #+
576 BNE #+
577 BNE #+
578 BNE #+
579 BNE #+
580 BNE #+
581 BNE #+
582 BNE #+
583 BNE #+
584 BNE #+
585 BNE #+
586 BNE #+
587 BNE #+
588 BNE #+
589 BNE #+
590 BNE #+
591 BNE #+
592 BNE #+
593 BNE #+
594 IM5
595 BNE IM5
596 AND #3
597 CMP #3
598 BCC #+1
599 RTS
600 DEX
601 BNE IM5
602 AND #3
603 CMP #3
604 BCC #+1
605 RTS
606 DEX
607 BNE IM5
608 AND #3
609 CMP #3
610 BCC #+1

```


Starship VC-20

611IM3	LDY #0	672	LDA (88),Y
612	LDA #FE	673	STA (90),Y
613	SBC #0	674	TYA
614	STA 88	675	BNE MD1
615	LDA #FF	676	LDA #32
616	SBC #0	677	LDY #31
617	STA 89	678	STA (88),Y
618IM6	LDA (##FE),Y	679	DEY
619	PLA	680	BPL *-5
620	LDA #32	681	LDA 89
621	STA (##FE),Y	682	CLC
622	PLA	683	ADC #132
623	CMP #54	684	STA 89
624	BCC IM61	685	LDA 91
625	EOB #2	686	ADC #132
626	STA (88),Y	687	STA 91
627IM61	INY	688	LDY #128
628	CPY #51	689	DEY
629	BEQ IM6	690	LDA (88),Y
630	PLP	691	STA (88),Y
631	BRQ **1	692	TXN
632	RTS	693	BRW #0
633	LDY #0	694	CLL #0
634	STA 87	695IM13	LDA (7),A
635	LDY INY+1	696	CPY #04
636	LDA INY	697	BCC #B4
637	LDY 1FF	698	CPY #00
638	STA #FE	699	BCC MDS
639	LDY #0	700IMD4	LDX #0
640IM1	LDA (##7),Y	701	BEQ #28
641	CPY #54	702	STX 3
642	BEQ IM6	703IMD3	LDY 3
643	PLP #1	704	RTS
644	SWR 87	705IM1	LDA #C
645	STA 87	706	BEQ #0
646IM6	LDA #FE	707	BEQ #07
647	CLC	708	BRQ **1
648	ADC #44	709IMB1	RTS
649	STA #FE	710	LDA #0
650	BEQ #31	711	STA MD1
651	LDY #FF	712	LDY #B8
652	BEQ #	713	BNE #B2
653	SWR INY	714	LDA #22
654	LDA 87	715	LDA #7C
655	AND #0	716IMB3	STA BLA+1
656	BEQ **1	717	DEX
657	RTS	718	BRW #B3
658	INC #0	719	STA 18F
659IM1	LDA INY+1	720	DEC #2B
660	STA 88	721	BNE **2
661	LDY	722	INC 3
662	BEQ #22	723	RTS
663	STA 89	724IMB2	LDA BF
664	STA INY+1	725	LSP
665	LDA INY+1	726	LSP
666	STA 89	727	TRX
667	ADC #0	728	LDA BLA,X
668	STA INY+1	729	CPY #132
669	STA 91	730	BCC **4
670	LDY #198	731	SBC #8
671MD1	DEY	732	BOS **2

Starship VC-20

```

733      ADC #8
734      STA BLA,X
735      CLC
736      ADC #1
737      STA BLA+1,X
738      RTS
739HIBL  LDA HIMZ
740      BEQ HI1
741      DEC SBDZ
742      BEQ **+1
743HI1   RTS
744      LDA #8
745      STA SBDZ
746      DEC HIMZ
747      BEQ HI2
748      LDA SSCHADR+1
749      CLC
750      ADC #132
751      STA $FF
752      LDA SSCHADR
753      STA $FE
754      LDY #0
755      LDA ($FE),Y
756      EOR #32
757      STA ($FE),Y
758      INY
759      STA ($FE),Y
760      RTS
761HIY   LDY #0
762      LDA #32
763      STA (SSCHADR),Y
764      INY
765      STA (SSCHADR),Y
766      RTS
767ROBL  LDA BOBZ
768      BEQ BO1
769      DEC BDC
770      BEQ **+1
771BO1   RTS
772      LDA #3
773      STA BDC
774      DEC BOBZ
775      BEQ BOZ
776      LDA ISCHADR+1
777      CLC
778      ADC #132
779      STA $FF
780      LDA ISCHADR
781      STA $FE
782      LDY #0
783      LDA ($FE),Y
784      EOR #22
785      STA ($FE),Y
786      INY
787      STA ($FE),Y
788      RTS
789BOZ   LDY #0
790      LDA #32
791      STA (ISCHADR),Y
792      INY
793      STA (ISCHADR),Y

```

```

794      RTS
795IBL   LDA IBZ
796      BEQ **+5
797      DEC SBDZ
798      BEQ **+1
799      RTS
800      LDA #5
801      STA SBDZ
802      DEC IBZ
803      BEQ IB1
804      LDY #0
805      LDA (SSCHADR),Y
806      CMP #108
807      LDA B1
808      BCS IB2
809      AND #2
810      ADC #108
811IB3   STA (SSCHADR),Y
812      ADC #1
813      INY
814      STA (SSCHADR),Y
815      RTS
816IB2   AND ##FE
817      CLC
818      RNE IB3
819IB1   LDY #0
820      LDA #32
821      STA (SSCHADR),Y
822      INY
823      STA (SSCHADR),Y
824      LDA B1
825      CMP #68
826      BCS **+4
827      LDA ##10
828      BAE JSO1
829      CMP #12
830      JCS **+4
831      LDA ##C
832      RNE JSO1
833      LDA #2
834JSO1  JSR SOO
835      DEC AZI
836      BEQ **+1
837      RTS
838      LDA #48
839      STA AZI
840      LDA INVI
841      CLC
842      ADC #22
843      STA INVI
844      STA INVA
845      BCC **+6
846      INC INVI+1
847      INC INVA+1
848      INC AZB
849      LDA INVI
850      STA $FE
851      LDA INVI+1
852      CLC
853      ADC #132
854      STA $FF

```

Starship VC-20

```
855 LDA #F1
856 LDY #43
857 STA ($FE),Y
858 DEY
859 BPL *-5
860 LDA $FE
861 CLC
862 ADC #44
863 STA $FE
864 BCC #+2
865 INC $FF
866 LDA #F2
867 LDY #87
868 STA ($FE),Y
869 DEY
870 BPL *-5
871 LDA $FE
872 CLC
873 ADC #88
874 STA $FE
875 BCC #+2
876 INC $FF
877 LDY #220
878 LDA #F3
879 DEY
880 STA ($FE),Y
881 TFR
882 BPT *-8
883 LDA #0
884 STA LIZ
885 STA MR
886 LDA INVA
887 STA $FE
888 LDA INVA+1
889 STA $FF
890 LDA #64
891 LDY #14
892 STA ($FE),Y
893 DEY
894 DEY
895 BPL *-6
896 LDA #65
897 LDY #15
898 STA ($FE),Y
899 DEY
900 DEY
901 BPL *-6
902 LDA $FE
903 CLC
904 ADC #44
905 STA $FE
906 BCC #+2
907 INC $FF
908 LDA #68
909 LDY #14
910 STA ($FE),Y
911 DEY
912 DEY
913 BPL *-6
914 LDA #69
915 LDY #15
```

```
916 STA ($FE),Y
917 DEY
918 DEY
919 BPL *-6
920 LDA $FE
921 CLC
922 ADC #44
923 STA $FE
924 BCC #+2
925 INC $FF
926 LDA #68
927 LDY #14
928 STA ($FE),Y
929 DEY
930 DEY
931 BPL *-6
932 LDA #69
933 LDY #15
934 STA ($FE),Y
935 DEY
936 DEY
937 BPL *-6
938 LDA $FE
939 CLC
940 ADC #44
941 STA $FE
942 BCC #+2
943 INC $FF
944 LDA #72
945 LDY #14
946 STA ($FE),Y
947 DEY
948 DEY
949 BPL *-6
950 LDA #73
951 LDY #15
952 STA ($FE),Y
953 DEY
954 DEY
955 BPL *-6
956 LDA $FE
957 CLC
958 ADC #44
959 STA $FE
960 BCC #+2
961 INC $FF
962 LDA #72
963 LDY #14
964 STA ($FE),Y
965 DEY
966 DEY
967 BPL *-6
968 LDA #73
969 LDY #15
970 STA ($FE),Y
971 DEY
972 DEY
973 BPL *-6
974 RTS
```


Commodore & Sinclair

VC-20 · VC-64 · 4032 · 8032

ZX SPECTRUM

HARDWARE

VC-20 16K-RAM-Modul **149.-**
32K-RAM-Modul schaltbar **199.-**
MODULBOX mit 3 Steckplätzen **99.-**
EPROM-Modul inkl. Prommer **149.-**
IEC-BUS-Modul inkl. Kabel **249.-**

SPECTRUM JOYSTIX **69.-**
ermöglicht den Anschluß von
2 Atari- oder Commodore-Joysticks
an Ihren Spectrum. Einfache Abfrage
mit dem IN-Befehl.

SOFTWARE

GRANDMASTER

Das stärkste Schachprogramm der Welt
für Homecomputer!

Mit hochauflösender Grafik, viel
Bedienungskomfort und
unübertroffener Spielstärke.

NEU: ab sofort auch für CBM 4032 & 8032
erhältlich!

VC-20 (+ 8K), VC-64 **79.-**
4032 (mit 12"-Monitor) und
8032 inkl. HiRes-Eprom **99.-**

SPIELE für VC-20, VC-64 und
Spectrum in Maschinenspr. ab **19.-**

KINGSOFT

Alle Preise inkl. MwSt. 6 Monate Garantie.

Fordern Sie unseren Gesamtkatalog (2 DM) an!

Schnackebusch 4 · D-5106 Roetgen · Telefon 0 24 08/83 19

Editor für 6502-Systeme

Seine Reihe „Programmieren in Maschinensprache“ setzt Homecomputer diesmal fort mit dem Editor.

Sie haben nun eine Programmserie, mit der Sie Programme und Unterprogramme in der Assemblersprache für alle Microcomputer mit dem 6502-Prozessor schreiben können.

Obwohl die Reihe speziell für den VC-20 und VC-64 begonnen wurde, sind die Programme fast ohne Umschreiben auch für die CBMs, den Apple, den Atari, usw. verwendbar. Lediglich die Bildschirm-Befehle und Tastatur-POKES müssen angepaßt werden bzw. können einfach ignoriert werden.

Der Assembler, der in der April-Ausgabe abgedruckt wurde, hat früher als vorgesehen schon weitere Zusatzfunktionen erhalten, denen in den nächsten Monaten von Zeit zu Zeit weitere folgen werden.

Auch der Disassembler aus Heft 3, wo übrigens für den VC-20 zwei Zeilen beim Druck verlorengegangen sind, wurde entsprechend angepaßt. Man beginnt mit dem Einladen des Editors. Nach dem Starten erhält man eine Tabelle mit den Befehlen, die der Editor kennt und den möglichen Parametern.

Es genügt, wenn die inversen Buchstaben eingegeben werden und auch Blanks sind optional. Man kann aber auch den gesamten Befehl eingeben oder Teile davon, sofern die signifikanten Buchstaben am Anfang enthalten sind.

```

*50 PRINT"HOME COMPUTER-ASSEMBLER"
*100 READO$:READA$10$=O$+A$:READA$0$=O$+A$:DIMA$(500),A(500)
*110 FORI=0TO7:READI#:I:NEXT
*120 DIMO$(50):FORI=0TO55:READO$(I):NEXT
*130 FORI=0TO7:READS#:I:NEXT
*140 DIMA$(40):FORI=0TO31:READA$(I):NEXT
*200 PRINT"CASSETTE / DISK ADDRESS" :WAIT 100:SETG$=I$<"C"ORG$>"I"THEN200
*310 DI=0$="D":IFNOTDITHENGOSUB5000
*320 INPUT"FILENAME":F$:IFDITHENF$=F$+1:GOTO
*500 OPEN1,1+(7ANDDI),2ANDDI,F#:L=0
*510 L=L+1:INPUT#1,L$:FORI=1TO9:IFMID$(L$,I,1)<">":THENNEXT
*520 IFI=1THENA$(LC)=LEFT$(L$,I-1):A(LC)=PC:LC=LC+1
*530 PRINTRIGHT$(L$," "+STR$(PC),5)LEFT$(L$,I)TAB(10)MID$(L$,I+1)
*540 GOSUB5000
*540 PC=PC+L
*590 IF(64ANDST)=0THEN510
*610 CLOSE1
*1000 IFNOTDITHENGOSUB5000
*1010 OPEN1,1+(7ANDDI),2ANDDI,F#:F=-1:Z=0
*1020 Z=Z+1:INPUT#1,L$:FORI=1TO9:IFMID$(L$,I,1)<">":THENNEXT
*1030 GOSUB4000:IFC1=5STHEN4100
*1040 O=0:G1=1:IFDITHENG1000
*1050 IFAI=1AND(ORND10)PC=3THENC=0+8
*1060 O=O+(50-A)
*1070 IFL=1THENGOSUB5000:IFC1=4AND(ORND10)STHENC=5-PC-2:G=5+(256ANDS<0)
*1080 SW$=Z:Z0$=S=-20$+S:R=INTPC/256:V=INTPC/128:STAB$(17,0)
*1090 POKEPV+D:POKEV+1,S:IFL=3THENPOKEV+2,C1
*1100 PC=PC+L:IF(64ANDST)=LTHE+1000
*1110 CLOSE1
*1120 GOTO2000
*1150 IFOI=57THEN1100
*1160 K=0:FORP=PC+L-1:A$="":FORK=K+1TOLEN(L$):J$=MID$(L$,K,1)
*1170 IFJ$<"",THENA$=A$+J$:NEXT
*1180 GOSUB5000:POKEP,S:NEXTP:GOTO1100
*4000 A$=MID$(L$,I+1,3):L=0
*4010 FORJ=1TO1758STEP3:IFAS$>MID$(O$,J,3)THENNEXT:PRINT"OPCODE ERROR IN"Z:GOTO2000
*4020 OI=(J-1)/3:L$=MID$(L$,I+5)
*4030 IFOI=56ANDNOTFTHENA$=L$:GOSUB5000:A(LC-1)=S
*4032 IFOI=57THENL=1:FORJ=1TOLEN(L$):L=L-(MID$(L$,J,1)="") :NEXT
*4035 IFOI=58THENA$=L$:GOSUB5000:PC=S
*4040 IFOI=55THENRETURN
*4045 IFOI>21THENA$=L$:GOTO4135
*4050 IFLEN(L$)=0ORL$="A"THENB$="A":GOTO4140
*4060 A$="":B$="":FORI=1TOLEN(L$):J$=MID$(L$,I,1)
*4070 IFJ$="#":ORJ$="("THENB$=B$+J$:GOTO4130
*4080 IFJ$=")":ORJ$=")"THEN4130
*4100 IFJ$=","THENI=I+1:B$=B$+MID$(L$,I,1):GOTO4130
*4110 IFJ$=")",THEN4135
*4120 A$=A$+J$
*4130 NEXT
*4135 L=-<(OI>21)-<(OI>46)-<(OI>54):IFLTHENRETURN
*4140 FORAI=0TO7:IFB$<>AI#(AI)THENNEXT:GOTO19990
*4150 L=1-(AI>0)-(AI>3):IF(OI=12OROI=15)ANDAI=4THENAI=7
*4152 IF<2(AIANDAI<OI)>=0THEN19990
*4155 IFAI<6THENRETURN
*4160 GOSUB5000:IFE=10ROI=7THENL=3:RETURN
*4170 L=2-(S>255)
*4175 IFAI=7THENIFL=3AND(OI=10OROI=12)THEN19990
*4180 RETURN

```


Starship VC-20

```

*5000 FORI=1TOLEN(A$)*J=ASC(MID$(A$,I,1))
*5010 IFJ=43THENSAS=MID$(A$,I+1):A$=LEFT$(A$,I-1):GOSUB5040:SA=S A$=SA$ GOSUB5040
  S=S+SA:RETURN
*5020 IFJ=45THENSAS=MID$(A$,I+1):A$=LEFT$(A$,I-1):GOSUB5040:SA=S A$=SA$ GOSUB5040
  S=SA-S:RETURN
*5030 NEXT
*5040 IFASC(A$)=62THENS$=MID$(A$,2):GOSUB6000:SS=S/256:S=S-256*S%:RETURN
*5050 IFASC(A$)=63THENS$=MID$(A$,2):GOSUB6000:SS=INT(S/256):RETURN
*6000 E=0: S=VAL(A$):IFASC(A$)=48THENRETURN
*7010 T=ASC(A$):IFJ=36THENS$=0
*8000 FORJ=2TOLEN(A$):J$=MID$(A$,J,1):S=16*S+VAL(J$)+(ASC(J$)-55ANDJ$)*"0":NEXT:R
  ETURN
*9000 IFJ=37THENFORJ=2TOLEN(A$):S=S+S+VAL(MID$(A$,J,1)):NEXT:RETURN
*9010 IFJ=42THENS=PC+L:RETURN
*9020 FORI=0TOC-1:IFA$(A$(I))THENS=A(I):RETURN
*9030 NEXT: E=1:RETURN
10000 DATADBASELISTANDROLEORLSRJMPCDFORSTYSTASTSLD,FLALICPYCMPDECCPXSBCINC
10010 DATABRPHPLCALPASCPTIPHACLIRTSPLASACCT,ALPAC,TA,THALLVTSKINVDKLDJNKX
  N
10020 DATASDBFLMIBVOCBVBFCBDBNBERADTSPECUDFDRG
10030 DATA#(0,1,5,9),0,1,1
10040 DATA#(6,56,37,38,69,70,68,101,102,130,103,134,164,165,166,196,197,198,220,
  222
10050 DATA#200,0,6,24,40,56,64,72,88,96,104,100,107,113,152,154,168,170,184,186,2
  20
10060 DATA#202,216,232,236,248,16,46,80,112,144,170,200,240,32
10070 DATA#-4,-4,12,12,32,8,16
10080 DATA#22,193,64,200,193,222,193,96,222,193,100,222,194,194,222,194,66,222,2
  20,66
10090 DATA#222,204
10100 PRINT"ADP ERROR IN:"
10110 PRINT"NEILADEN"
10120 PRINT"EDITOR"
10130 PRINT"DISASSEMBLER"
10140 PRINT"
10150 PRINT"
10160 PRINT"
10170 PRINT"
10180 PRINT"
10190 INPUTA$:A$="X"+A$:GOSUB6000:PRINTS:GOTO6000
  REPT,

```

Die Zeilennummern mit Punkt wurden gegenüber der Heft 4/83-Version geändert. Für Kassettenbetrieb entfallen in den Zeilen 20005 und 20010 die letzten beiden Zeichen.

ADD
Um ein Assembler-Programm einzugeben, benutzt man den Befehl A(DD) (Anm.: die Buchstaben in Klammern sind optional). Dann erscheint eine "1" mit einem inversen Blank dahinter. Dies ist der Cursor, während die "1" die laufende Zeilennummer des Befehls ist. Wenn man einmal einen Befehl unter einer bestimmten Zeilennummer eingegeben hat, so heißt dies nicht, daß zu dieser Zeilennummer immer der eingegebene Befehl gehört, denn man kann auch noch nachträglich Befehle einfügen, und dabei ändern sich dann entsprechend die Nummern der folgenden Befehle.

ADD heißt immer Anhängen von weiteren Befehlen. Wenn der Speicher leer ist, dann lautet die „nächste“ Zeile 1, wenn bereits 43 Zeilen vorhanden sind, dann ist 44 die Zeile, in die nach ADD geschrieben wird.

INSERT
Hat man irgendwo im Assembler-Listing einen oder mehrere Befehle vergessen und nachzutragen, so benutzt man den Befehl I(NSERT). Und zwar gibt man nach dem Befehlswort die Zeilennummer an, hinter der die Befehle eingeschoben werden sollen. Fehlen Befehle direkt am Anfang, so wird nur der Befehl ohne Parameter eingegeben und die Zeilen werden solange eingeschoben,

bis eine neue Editoranweisung dieses beendet.

DELETE
Sind Assembler-Zeilen überflüssig geworden, so kann man sie mit D(ELETE) entfernen. Wie bei dem BASIC-Befehl LIST kann man hinter dem Delete-Befehl entweder nur eine Zeilennummer eingeben, wenn man nur eine Zeile löschen will; oder z. B. "D-5" eingeben, wenn man die ersten 5 Zeilen löschen möchte; oder "D5-", wenn alle Zeilen ab Nummer 5 gelöscht werden sollen; oder einfach "D", um sämtliche Zeilen zu löschen. Damit man nicht aus versehen Zeilen entfernt, die man noch benötigt, wird zur Sicherheit noch einmal

gefragt, ob man auch sicher ist. Die Frage (in Englisch) taucht immer dann auf, wenn mehr als 10 Zeilen gelöscht werden sollen.

LIST
Auch bei L(IST) ist die Parameterübergabe wie im BASIC. Wird beim Listen die Taste "S" gedrückt, so stoppt der Editor und erwartet zur Fortsetzung einen weiteren Tastendruck. Drücken der Taste "Q" beendet das Listen bevor die letzte Zeilennummer erreicht ist.

EDIT
Stellt man fest, daß eine Zeile falsch eingegeben wurde, so kann sie mit dem E(DIT)-Befehl korrigiert werden. Die Parameterübergabe ist wie unter D(ELETE) beschrieben. Der Editor listet die zu korrigierende Zeile und setzt den Cursor an den Zeilenanfang.

Diesmal ist es ein „echter Cursor“ und es zählt, wie vom CBM gewohnt, alles, was in der oder den entsprechenden Bildschirmzeilen steht. Deshalb kann man nach Durchführen der Korrektur sofort "RETURN" drücken, und auch der Rest nach dem Cursor wird übernommen. Oder nur "RETURN", wenn in der Zeile nichts geändert werden soll.

Auch bei dem EDIT-Befehl kann man einfach eine neue Editor-Anweisung eingeben, wenn man aufhören will, sofern man den Rest der Zeile (oder Zeilen) mit Spaces löscht.

COPY
Wenn man einen Block, den man schon einmal eingegeben hat, an einer anderen Stelle haben möchte, so verwendet man den Befehl C(OPY). Direkt hinter dem Befehl gibt man aber noch keine Parameter an. Der Editor fragt diese näm-

Starship VC-20

```
80 POKE45,32:POKE46,31
90 CLR
100 DIMA$(500)
110 L$=" ":FORI=1TO5:L$=LEFT$(L$+L$,21):NEXT:L$=L$+" "
120 PRINTTAB(6)"HOME COMPUTER
200 PRINT"HOME COMPUTER EDITOR
210 PRINT"APPEND [A]
215 PRINT"SAVE [A][~][E]
220 PRINT"VERIFY [A][~][E]
225 PRINT"ADD
230 PRINT"INSERT [A]
235 PRINT"EDIT [A][~][E]
240 PRINT"DELETE [A][~][E]
245 PRINT"LIST [A][~][E]
250 PRINT"COPY
255 PRINT"QUIT -> ASS + DISASS
280 IFB=1ORB=5ORB=9THENA$="L":GOTO300
290 INPUT"BEFEHL";A$
300 PRINT:RESTORE:FORB=1TO10:READB$:IFLEFT$(A$,LEN(B$))<>
B$THENNEXT
310 IFB<>2ANDB<9THENBOSUB5000
330 ONBGO TO400,600,500,700,800,1000,1200,900,1100,540,200
400 REM APPEND
410 GOSUB6000
420 INPUT#1,A$
430 IFN2<NTHENFORC=NTO N2+1STEP-1:A$(C+1)=A$(C):NEXT
440 N2=N2+1:N=N+1:A$(N2)=A$:IF(64AND$T)=0THEN420
450 CLOSE1:GOTO280
500 REM SAVE
502 INPUT"FILENAME";F$:PRINT CASSETTE / DISK  WAIT
198,1:GETG$:IFG$="C"THEN510
505 IFG$<>"D"THEN502
507 OPEN1,8,2,F$+".S.W":GOTO520
510 OPEN1,1,1,F$
520 FORC=N1TON2:PRINT#1,CHR$(34)A$(C):NEXT:CLOSE1
540 PRINT"ASSEMBLER EINLADEN ?
550 WAIT198,3:GETG$
560 IFG$="J"THENCLR:LOAD"ASSEMBLER",8
570 PRINT"DISASSEMBLER ?
580 WAIT198,3:GETG$
590 IFG$="J"THENCLR:LOAD"DISASSEMBLER",8
595 GOTO200
600 REM ADD
610 C=N:PRINT"
620 GOSUB4000:IFBFTHEN300
630 PRINTL$:GOSUB4500
640 A$(C)=A$:N=C:GOTO620
700 REM LIST
710 PRINT"":FORC=N1TON2:A$=A$(C):GOSUB4500
720 GETG$:IFG$="S"THENWAIT198,1:GETG$
730 IFG$="Q"THEN280
740 NEXT:GOTO280
800 REM DELETE
810 D=N2-N1+1:IFD>10THENPRINT"ARE YOU SURE ?  WAIT198
,7:GETG$:IFG$="N"THEN280
820 N=N-D
830 FORC=N1TON:A$(C)=A$(C+D):NEXT:GOTO280
900 REM VERIFY
910 GOSUB6200
920 FORC=N1TON2:INPUT#1,A$
930 IFA$(C)>A$(C)OR64AND$TANDL<N2THENPRINT"VERIFY ERROR IN
C:CLOSE1:GOTO280
940 NEXT:CLOSE1:PRINT"OK":GOTO280
```

lich der Reihe nach ab. Er beginnt mit dem Bereich der zu kopierenden Zeilen. Die Zeilennummern werden wie bei DELETE, LIST oder EDIT behandelt.

Danach erfolgt nach Anweisung die Eingabe der Zeilennummer, hinter der die Zeilen eingeschoben werden sollen (fast wie bei INSERT).

Man kann den Block nämlich nicht nur anhängen, sondern an jede Stelle innerhalb des Assembler-Listings einfügen, sogar innerhalb des Blocks selbst. Lediglich wenn man den Block direkt an den Anfang kopieren möchte muß man sich eines kleinen Tricks behelfen. Man fügt dann nämlich vorher einfach eine neue Zeile 1 ein, kopiert den Block (Vorsicht! Die zu kopierenden Zeilennummern haben sich jetzt um 1 verschoben!) hinter die Zeile 1 und löscht dieselbe mit D1. Der zu kopierende Block steht jetzt am Anfang.

APPEND SAVE VERIFY

Drei Befehle dienen dem Laden, Retten und Verifizieren von Assembler-Files auf Diskette oder Kasette.

Der Befehl LOAD, den man in diesem Zusammenhang erwartet gibt es nicht. Ihn ersetzt der Befehl AP(PEND), der einen Parameter haben kann; die Zeilennummer, hinter der das File vom Datenträger in das im RAM befindliche Programm geladen werden soll.

Ist der Speicher leer, so wird sich nach APPEND nur das Programm, das zugelesen wurde, im Speicher befinden.

APPEND ohne Parameter hängt das Programm an das im Speicher befindliche an!

Um das Assembler-Listing auf Kasette oder Disk auf-

Starship VC-20

zunehmen, wird der Befehl S(AVE) benutzt. Ohne Parameter speichert man das gesamte, im Speicher befindliche Programm; und mit entsprechenden Parametern (wie oben unter DELETE beschrieben) können Teile des Programms übertragen werden.

Um sich zu vergewissern, daß alles richtig gespeichert wurde, kann mit V(ERIFY) die Korrektheit der Aufnahme überprüft werden. Dabei ist zu beachten, daß wenn Parameter übergeben wurden, diese identisch sind mit denen, die beim Kommando SAVE verwendet wurden.

GOTO 280

Hat man aus irgend einem Grunde Stop oder Break gedrückt, so kann mit GOTO 280 der Programmablauf fortgesetzt werden, ohne daß das Assembler-Listing gelöscht wird (was ja nach RUN der Fall wäre).

QUIT

QU(IT) sorgt als Befehls-eingabe dafür, daß Assembler oder Disassembler ausgeführt werden.

Achtung! Zuerst muß das Assembler-Programm im Speicher auf einen Datenträger (Kassette oder Diskette) aufgenommen werden! Erst dann kann das Programm assembliert werden.

Ist der Assembler nachgeladen, dann werden wie vom Bildschirm her befohlen, die Kassette zurückgespult und die weiteren Befehle befolgt. Dies entfällt bei Diskettenbetrieb, wo lediglich darauf zu achten ist, daß sich Assembler und Assembler-Programm auf der selben Diskette befinden.

```

1000 REM EDIT
1010 FORC=N1T0N2 PRINT"QUIT" S=PEEK(214) A#=A#(C)-GOSUB4500
1015 PRINT"Q";IFPEEK(214)>5THEN1015
1020 A#="" :SYS50528
1030 FORI=515T0500 IFPEEK(I) THENA#=#A#+CHR$(PEEK(I))-NEXT
1040 GOSUB4100:IFBFTHEN300
1050 A#(C)=A# NEXTC GOTO280
1100 REM COPY
1110 INPUT"COPY-BEREICH"/A#
1120 GOSUB5000:INPUT"WOHIN (ZEILENNUMMER)"/C
1130 IFC>NTHENC=N
1140 D=N2-N1+1
1150 IFC<NTHENFORI=NT0C+1STEP-1 A#(I+D)=A#(I) NEXT
1160 N=N+D:IFN1>DTHENN1=N1+D
1170 FORI=C+1T0C+D:A#(I)=A#(N1):IFN1=CTHENN1=N1+D
1180 N1=N1+1:NEXT
1190 GOTO280
1200 REM INSERT
1210 C=N1
1220 GOSUB4000:IFBFTHEN300
1230 PRINTL# :GOSUB4500
1240 FORI=NT0CSTEP-1:A#(I+1)=A#(I):NEXT N=N+1 N=C+D=A#
1250 GOTO1230
4000 A#="" I=1 C=C+I C#=RIGHT$( " "+STR$(C),3)
4005 PRINT"QUIT"
4010 S=PEEK(214) PRINTC#LEFT$(A#,I-1) & LEFT$(MID$(A#,1,I)+ " ",1)'#
" MID$(A#,I+1) " Q
4015 IFPEEK(214)>5THENPRINT"Q" :GOTO4015
4020 WAIT198.15 GETB# :B=ASC(B#)
4030 IFJ=29THEN4030
4040 IFJ=157THENI=I+1(I=1) GOTO4010
4050 IFJ=148THENA#=LEFT$(A#,I-1) +MID$(A#,1,I)GOTO4010
4060 IFJ=20THENA#=LEFT$(A#,J-1) +MID$(A#,J) :I=I+1(I=1) GOTO4010
4070 IFJ=13THEN4100
4080 A#=LEFT$(A#,I-1)+C#+MID$(A#,I+1)
4090 I=I+1(I=LEN(A#)) GOTO4010
4100 FORI=1T0LEN(A#) :IFMID$(A#,I,1)<>" " THENNEXT BF=1 RETURN
4110 IFMID$(A#,I+1,1) = " " THENA=LEFT$(A#,I)+104*(A# I=2) GOTO4100
4120 BF=0 RETURN
4500 PRINTRIGHT$( " "+STR$(C),3)
4510 FORI=1T08 I#=#MID$(A#,I,C)+I-1 I#I# THENNEXT I=I-1
4520 PRINTAB"13" MID$(A#,I,C) :RETURN
5000 D=B FORI=1T09 J=ASC(MID$(A#,I,1)+ " ") IFJ=4 THEND=1
5010 IFJ<4508 J=17 THENNEXT
5020 N1=VAL(MID$(A#,I))
5030 N2=VAL(MID$(A#,I+LEN(I#)+1))
5040 IFN1=0 THENN1=1:MID$(A#,I+1)
5050 IFD THENN2=N1+N2
5060 IFRIGHT$(A#,2) = THENN2=N
5070 IFRIGHT$(A#,2) = THENN2=N
5080 IFN2<N THENN2=N
5090 IFN1<N THENN1=N
5100 RETURN
6000 INPUT"FILENAME" :E# PRINT"CHECKING" :DISK#=# :WAIT198.1 A#
T0# :IFC#>D THEN6100
6002 IFC#>D THEN6000
6005 PRINT"CASSETTE" :A#=# :A#=#
6010 PRINT"DISK" :A#=# :A#=# :PRINT"ASSEMBLER FILES"
6020 PRINT"DIR" & P# :DRU#=#
6030 WAIT198.1 GETB# PRINT"OPEN" :I=C :E# RETURN
6100 OPEN"13" :A#=# :E# RETURN
10000 DATA#A# :S# :I# :I# :I#

```

READY.

Für Kassettenbetrieb entfallen in den Zeilen 560 und 590 die letzten beiden Zeichen.

SYNTAX und Features

Gibt man direkt hinter einer Zeilennummer einen Editorbefehl ein, so wird dieser ausgeführt - und nicht in die Zeile übernommen!

Alles andere, was direkt hinter einer Zeilennummer steht und nach einem Blank mit weiteren Eingaben versehen ist, wird in die entsprechende Zeile geschrieben.

Labels werden direkt hinter die Zeilennummer geschrieben und dürfen keine Blanks enthalten. Ein Blank, das einem Label folgt, bedeutet dem Editor, daß nun ein Mnemonic-Befehl folgt. (Mnemonics, soviel sei für Anfänger gesagt, nennt man die Codewörter wie z. B. LDA (Load Akku), STA (Store Akku), TAX (Transfer Akku to X-Register), usw. - unser 6502-Lehrgang, der in Kürze beginnt und zu dessen

Zweck die Programme vorgestellt wurden, wird darüber Auskunft geben)

Folgt direkt hinter der Zeilennummer ein Blank dann gehört hinter dieses Blank ein Mnemonic.

Dahinter gehört, wenn es sich um einen Mehrbyte-Befehl handelt, wieder ein Blank und dann ein Operand.

Ein Semikolon wird Kommentaren, die zur besseren Verständlichkeit des Programms einer Programmzeile beigefügt werden können, vorangestellt.

Der Assembler versteht Hex- und Dezimalzahlen. Hexzahlen haben ein Dollarzeichen vorangestellt.

Direkte und relative Sprünge zu Labels und zu labelabhängigen Punkten sind möglich. Z. B.:

```
JSR LABEL
oder
JSR LABEL+86
BNE MARKES8-9
```

Auch Sprünge relativ zum Programm-Counter sind gestattet. Das Sternchen "*" ist in diesem Fall das

Symbol für den Programm-Counter. Beispiel: BNE. -3 Die bisher zusätzlich vom Assembler verstandenen Mnemonics sind:

ORG Programmbeginn

EQU (Equate)

DFB (Define Byte)

ORG hat kein Label und muß eine Hex- oder Dezimalzahl im Operandenfeld haben.

Vor EQU muß sich ein Label befinden und dahinter eine entsprechende Hex-, Dual- oder Dezimalzahl.

Vor DFB kann ein Label stehen. Dem Mnemonic folgen ein oder mehrere Hex-Dual- oder Dezimalzahlen. Mehrere Zahlen werden durch Kommata getrennt.

Eine weitere nachträglich eingeführte Bequemlichkeit ist, daß mit den Größer-als-und Kleiner-als-Zeichen Labels in High- und Low-Byte zerlegt werden können.

Z. B.

```
1ADRESSE EQU 34356
2      LDA #<ADRESSE
3      STA HIGHBYTE
4      LDA #>ADRESSE
5      STA LOWBYTE
```

Wie bereits angesprochen, werden weitere „Bequemlichkeiten“, nützliche Befehle und Hilfsprogramme

zum Programmieren in Assembler (und somit in Maschinencode) folgen.

Änderungen gegenüber Disassembler Programm aus Heft 3/83

```
600 INPUT"###":A$ IF A$="E" THEN CLR LEND:EDITOR"AS
605 IF A$="L" THEN 1000
610 IF RIGHT$(A$,1)="/" THEN GOTO 600:GOTO 605
620 GOTO 600
630 IF A$="C" THEN A$="C" THEN 610
640 GOTO 600:GOTO 605:RETURN
650 PC=0:FOR I=2 TO LEN A$:A$(I)=I
660 B=ASC(MID$(A$,I,1)):IF B=ASC("A") THEN B=B-7
670 B=ASC(MID$(A$,I,1)):IF B=ASC("a") THEN B=B-7
680 FOR J=STOBYTE+1 TO INT(LEN(B)/16)+1:DC=DC+B*16:J: A$=A$+B:IF A$? THEN A$=A$+7
690 PRINT CLR CRT:GOTO 605
700 FOR J=DC TO DC+B-1:IF INT(J/16)=INT(DC/16)+1: A$=A$+B:IF A$? THEN A$=A$+7
710 PRINT CHR$(J):NEXT J:RETURN
1000 FOR I=1 TO LEN A$:DC=PC:PRINT "###":GOTO 600:PRINT "###":A$=PEEK(PC)
1010 CLR:PC=PC+1:GOTO 600:GOTO 605:GOTO 600
1020 GOTO 600
1100 DC=P1:GOTO 600:PRINT "###":DC=DC+1:PC=PC+1:RETURN
1200 DC=P1:GOTO 600:PRINT "###":P2=PEEK(PC+1):DC=P2:GOTO 600
1210 PRINT "###":TAB(10):DC=P1
1220 IF P1? THEN P2=PEEK(PC+1):IF P2(128)=P2 AND P2=2+2:GOTO 1220
1230 IF P1? THEN P2=PEEK(PC+2):P2=255-P2+1
1240 PRINT "###":IF P1? THEN P2=PEEK(PC+3):IF P1(128)=P2 AND P2=9:GOTO 1240
1250 GOTO 600
1260 PRINT TAB(20):PC=P1:RETURN
1300 DC=P1:GOTO 600:PRINT "###":P2=PEEK(PC+1):DC=P2:GOTO 600:PRINT "###"
1310 PC=PEEK(PC+2):DC=P2:GOTO 600:PRINT "###":TAB(10):M$(P1)
1320 PRINT "###":IF P1? THEN P2=PEEK(PC+3):GOTO 600:PRINT P2$(P1):PC=PC+3:RETURN
###:###
```

Zeile 602 entfällt

Zeile 1020 entfällt

Für Kassettenbetrieb entfallen in Zeile 600 die letzten zwei Zeichen.

Starship VC-20

Labyrinth in 3,5 K!

Wieviele Leser von Homecomputer uns angerufen haben, haben wir nicht gezählt, aber wenn es um das Labyrinth für den VC-20 ging, dann lag die Antwort auf der Hand, keine Speichererweiterung vorhan-

den, Programm also nicht lauffähig. Wir haben uns deshalb bemüht (vielen Anrufern habe ich es versprochen), falls es überhaupt möglich ist, das Programm für den VC-20 ohne Erweiterung umzuschrei-

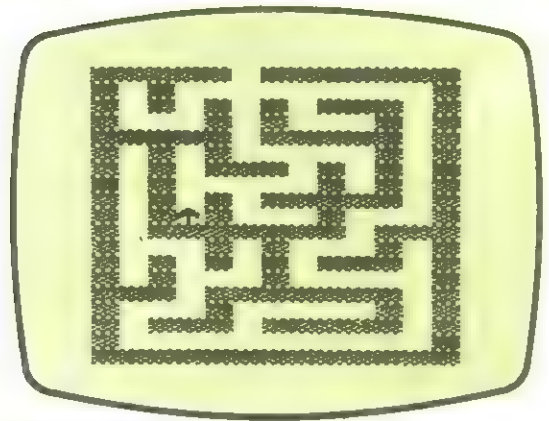
ben, es zum Laufen zu bringen.

Am Ende konnte durch Renummerierung, Dimensionierung, Streichen der Texte und Poken eines neuen RAMTOPs das Programm so weit gekürzt wer-

den, daß es inclusive Variablen fast genau soviel Platz verbraucht, wie ohne Speichererweiterung zur Verfügung stehen.

```
1 DIMM(7),W(3)
2 POKE52,29:POKE34,29:POKE36,29
3 VP=80726:CO=3
4 GOTO5
5 V=PEEK(TC-U):D=PEEK(TC-U+LL):IFABS(U)=1ORDC=0THEN7
6 IFV=32AND2=32THENFORA=U-U+LLTOBC-V-LLSTEPLL:POKEA,W3:NEXT
7 POKETC,TW:POKEBC,BW:POKETC+VP,CO:POKEBC+VP,CO
8 B=B+1:IFBC=DTHENTC=TC+U:BC=BC+V:GOTO7
9 IFABS(CO)=1THENPOKETC,T1:POKEBC,B1:POKETC+VP,CO:POKEBC+VP,CO
10 IF(TC=BC)ALLTHENRETURN
11 FORI=TC+LLTOBC-LLSTEPLL:POKEB,SW:POKEB+VP,I3:NEXT:RETURN
12 P=AL:U=LL+1:V=I-LL:T1=80:B1=122:SW=103
13 FORX=4TO8STEP2:T=V+1:D=V-U:B=0:IFX=4THENI3
14 C=NR:U=LL-1:V=LL-1:W=101:T1=7:R1=75:R=78:BW=77
15 U=NR+1:D=NR+1
16 IFV=32AND2=32DIMM(X)
17 T=LL+D:U=U+D
18 U=U-D*INT(U/D)+D:IFD=0THENTC=M(X):BC=M(X)+1:GOSUB5:GOTO21
19 U=99:BW=100:U=U-LL:V=V+LL:W=PEEK(TC+U)
20 IFN=D*INT(U/D)+D:IFD=0:TC=M(X)+D*LL:BC=M(X)+1+D*LL:GOTO33
21 IFABS(CO)=1THEND=U-LL:V=V+LL
22 M(X)=M(X)+D*W:U=U+D*W+1:POKEX+1,W-D*W+M(X)
23 IFV=32AND2=32THENRETURN
24 IFV=D*INT(CO/D)+D=ATHEM2
25 OP=OP+M(X):S=PTC+CO:CO=CO+1:IFDC<CDTHENM2
26 IFCO=CDTHENRETURN
27 R=PEEK(CO+4)+1
28 IFR=32THEN(R+(R+4)-1)/D5)=(R/3)+1:IF(R/4)<OTL+U*LLTHENM7
29 R=PEEK(CO+4)+1
30 IFR=32THENM6=(R/3)+1:M7=(R/3)+1:IFM(5)<OTR+D*LLTHENM9
31 FORTC=M4+TOM6:POKETC,W1:POKETC+VP,CO:NEXT
32 FORTC=M5+TOM7:POKEBC,W2:POKEBC+VP,CO:NEXT:RETURN
33 PRINT"OK'LL'
34 GOTO# IFV#="THEN34
35 PRINT
36 NP=P:G=PEEK(P):IFG=12:THENNP=P+M(X)
37 G=PEEK(P+M2)
38 IFR=32THENPRINT"ENLE":GOSUB2:RUN
39 T1=D*INT(CO/D)+D<0THENP=NP:PRINT"SIE GEHEN":GOTO48
40 IFRND(1)>.97THENPRINT"RA":OL=OL+1
41 PRINT"WARD GERAMMT":GOSUB1:GOTO31
```

Starship VC-20



```
42 NM=M: IFA$="R" THEN NM=M+1
43 IFA$="M" THEN GOSUB 67
44 IFA$="L" THEN NM=M+3
45 IF NM=M THEN 33
46 IF NM>3 THEN NM=NM-4*INT(NM/4)
47 M=NM: PRINT "DREHEN"
48 D=W(M): Q=PEEK(P): OP=P: ML=M-1: MR=M+1
49 IF ML<0 THEN ML=3
50 IF MR>3 THEN MR=0
51 OX=OX-OL: IFOX<0 THEN PRINT "LUFT=0": GOSUB 92: RUN
52 T=T+1: IFT=TT THEN TT=1: PRINT "LICHT": GOSUB 91
53 IFTT=1 THEN DC=0: GOSUB 90: GOSUB 12
54 IFF=MC AND RM=0 THEN RM=1: GOSUB 67
55 M(4)=TL: M(5)=BL: M(6)=TR: M(7)=BR: GOTO 33
56 P=P+1: IFF>EM THEN P=SM
57 A=INT(RND(1)*4): DC=0
58 A=A+1: DC=DC+1: IFC>3 THEN 56
59 IFA>3 THEN A=0
60 M=P+M(A): IF M<SM OR M>EM THEN 58
61 CP=PEEK(P): CM=PEEK(M): IFC>0 AND CP=210 THEN P=M: GOTO 57
62 TM=M-SM: IF (CP=CM OR CM<210) AND CO>0 THEN 58
63 ME=TM-L*INT(TM/L): IF (ME=0 AND M(A)=1) OR (ME=0 AND M(A)=-1) THEN 58
64 OD=INT(15/W(A)): CP=CP/W(A): POKE P, CP: CM=CM/OD: POKE M, CM
65 P=M: CO=C+1: IFC<H THEN 57
66 RETURN
67 GOSUB 90: PRINT "KARTE"
68 A=TL+LL+1: MS=A: DC=SM: DD=SM+6
69 FOR B=DC TO DD: FOR C=0 TO 3: CB=PEEK(B): CB=CB-W(C)*INT(CB/W(C))
70 AA=1: IFC=1 OR C=3 THEN AA=LL
71 IFC>1 THEN AA=-AA
72 BB=LL/AA: P2=A+AA: P1=P2+BB: P3=P2-BB
73 IF B=0 THEN POKE P1, W4: POKE P2, W5: CO
74 IF B=0 THEN POKE P1+W3: POKE P2+W3: W3: POKE P3+W3
75 IF B=0 THEN POKE P1+vP, CO: POKE P2+vP, CO: POKE P3+vP, CO
76 NEXT: A=A+2: NEXT: DC=DC+L: DD=DD+L: A=MS+L2*LL: MS=A
77 IF DD<EM THEN 69
78 GOSUB 91: RETURN
79 REM
80 TL=7680: BL=8164: TR=TL+21: BR=BL+21: LL=82: LW=3: CD=3
81 W1=99: W2=100: W3=160: W4=88
82 W(0)=5: W(1)=7: W(2)=3: W(3)=2: L=INT(RND(1)*3+6)
83 M(0)=1: M(1)=L: M(2)=-1: M(3)=-L: M(4)=TL: M(5)=BL: M(6)=TR: M(7)=BR
84 W=INT(RND(1)*5+6): H=L*W-1: B=L-1: C=0: DC=0: T=0: RM=0
85 SM=7480: EM=SM+H: FOR A=SM TO EM: POKE A, 210: NEXT: P=SM+INT(RND(1)*H)
86 EC=SM+INT(RND(1)*L): MC=INT(RND(1)*M)+SM
87 OX=INT(H/3)+1: IFC<0 THEN OX=-OX
88 OX=OX+H: OL=1: TT=INT(RND(1)*H/8)+1: GOSUB 55: GOSUB 96
89 M=0: NP=P: CE=PEEK(EC): CE=CE/W(3): POKE EC, CE: GOTO 48
90 PRINT "J": RETURN
91 FOR X=0 TO TL: NEXT: RETURN
92 GET A$: IFA$="" THEN 92
93 RETURN
```

READY.

IBM PERSONAL COMPUTER

6 Monate Garantie

Computerversand

Albert-Schweitzer

3436 Hessisch L.

Alle genannten Preise incl. Mehrwertsteuer



Systemeinheit 64 KB + Disklaufwerk 320 KB	6590,--
Tastatur	870,--
Bildschirm	1110,--
Adapter für Drucker und Bildschirm	1090,--
Grafik Drucker	1980,--
Druckerkabel	185,--
2. Laufwerk 320 KB	1495,--
DOS	137,50
Speichererweiterung	950,--
Komplettes System (wie vor)	13300,--

Systemeinheit 64 KB + Disklaufwerk 160 KB	5950,--
Tastatur	870,--
Bildschirm	1110,--
Adapter für Drucker und Bildschirm	1090,--
DOS	137,50
Komplettes System (wie vor)	8690,--



H. Schädel

...-Straße 1

...ichtenau

...rtsteuer + Versandkosten

olivetti - M 20 -

Der eingeführte bewährte Personal Computer mit den
perfekten Problemlösungen



olivetti-Computer M 20
mit 160 KB
+ 2 Disklaufwerken à 320 KB 9950,--



TI 99/4A

6 Monate Garantie

TI 99/4A 16 K RAM	699,--
Recorder	135,--
Kabel für Recorderanschluß	47,75
Sprachsynthesizer	349,--
Joysticks	95,--

Schach- programm Grandmaster - ein Meister seines Fachs

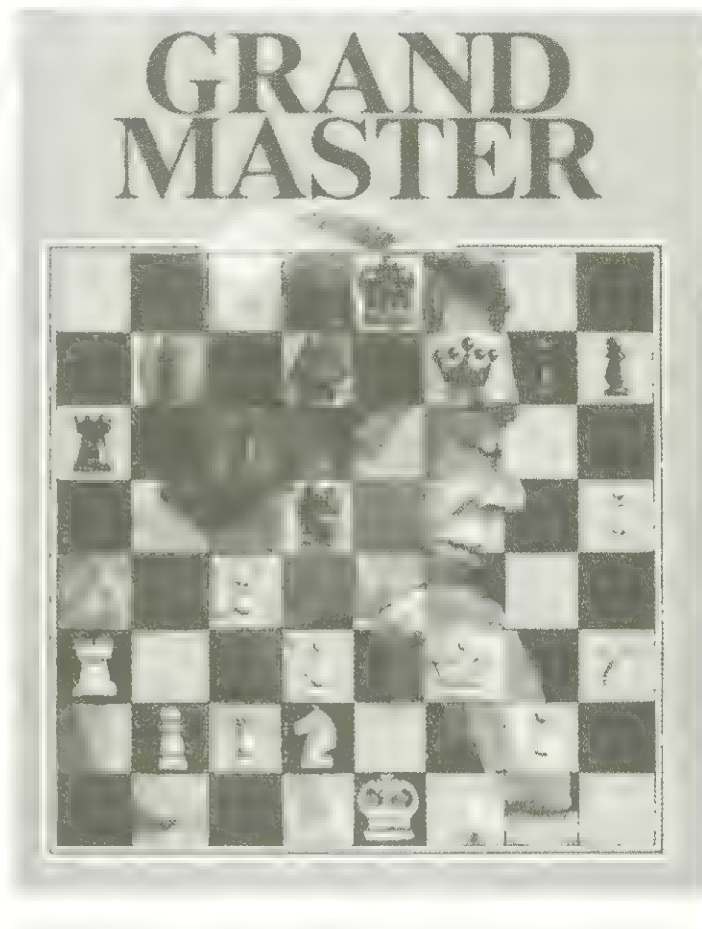
Seit es Computer gibt, übte das „Königliche Spiel“ eine starke Faszination und Herausforderung auf viele Leute aus, die sich mit dem Programmieren beschäftigen. War es anfangs selbst auf großen Anlagen kaum möglich, den Computer zu einem guten und für den geübten Spieler ernsthaften Gegner zu entwickeln, so ist dies heute dank der rasanten Entwicklung der Mikroprozessortechnik selbst bei den Kleinen möglich geworden.

Aus der Fülle der angebotenen Schachprogramme, die inzwischen auf dem Markt sind, fiel uns das 1982 von der Firma „Kingsoft“ entwickelte Programm GRANDMASTER auf. Grandmaster gibt es als Cassette für den VC-20 und VC-64. Für die VC-20 Version ist eine Speichererweiterung um mindestens 8 K notwendig.

Kingsoft selbst bezeichnet sein Programm als das „stärkste Schachprogramm für Heimcomputer“. Diese Behauptung möchten wir an dieser Stelle weder be-

stätigen noch in Frage stellen. Nachdem wir Grandmaster jedoch ausführlich getestet haben, können wir sagen, daß dieses Programm in punkto Handling, Grafik und Spielstärke sehr ansprechend und last not least auch vom Preis (79,- DM unverbindl.) recht interessant ist.

Die gutverständliche deutsche Bedienungsanleitung rundet das Programm ab. Nach dem Laden des Programms waren wir von der Grafik und der übersichtlichen Anordnung der Bildschirmanzeigen äußerst



angetan. Für die Ästhetiker unter den Schachspielern wurden einige reizvolle Möglichkeiten, den Bildschirm nach persönlichem Geschmack einzurichten, eingebaut. So kann die Grafik mittels der Cursor-tasten in horizontaler und vertikaler Richtung exakt auf die Bildschirmmitte zentriert werden. Über die Funktionstasten können das Spielbrett sowie Vorder- und Hintergrund nach Lust und Laune jede mögliche Farbkombination erhalten. Zugegeben, eine Spielerei, aber dennoch sehr schön!

Doch nun zum eigentlichen Spiel: Es stehen 10 Spielstufen mit Rechenzeiten zwi-

schen 5 Sekunden und 2 Stunden zur Verfügung. Für Freunde des Briefschachs ist Stufe 0 vorgesehen, bei welcher der Computer Stunden oder unter Umständen sogar Tage für seine Zugentscheidung benötigt.

Eine Änderung der Spielstärke ist zu jeder Zeit über die Taste „S“ in Verbindung mit der entsprechenden Zahl möglich.

Während Grandmaster überlegt, zeigt er den momentan favorisierten Zug und wieviel Halbzüge er im Voraus analysiert hat. Betätigen der „Stop-Taste“ unterbricht seine Überlegungen und veranlaßt ihn, diesen Zug auszuführen. Diese Funktion ist sehr praktisch, wenn auf einer

CAR-CRASH

höheren Spielstufe gespielt wird und der Spieler nicht die endgültige Entscheidung des Computers abwarten möchte.

Die folgenden Möglichkeiten ergänzen die praktische Handhabung. Mit der Funktionstaste F2 kann auf automatisches Spiel umgeschaltet werden. Grandmaster spielt nun so lange gegen sich selbst, bis er ein Matt oder Patt findet. Für einen Anfänger zum Erlernen von Strategien ideal geeignet! Einfaches Betätigen der STOP-Taste führt wieder zum normalen Spiel zurück.

Um eine Figur zu setzen, wird lediglich die Start- und Zieladresse eingegeben (z. B. E2 E4) und RETURN gedrückt. Mittels der DEL-Taste können falsch eingegebene Zeichen gelöscht werden, wenn RETURN noch nicht gedrückt war. Auch nach dem RETURN-Befehl ist eine Zugrücknahme noch möglich und zwar über die ←-Taste. Für den ernsthaften Spieler ist dies natürlich Mogelei und daher nicht akzeptabel; für Anfänger jedoch unter Umständen lehrreich.

Grandmaster verfügt selbstverständlich über eine Eröffnungsbibliothek, die alle gängigen Varianten der Spieleröffnung enthält. CTRL 0 dreht das Spielbrett um; der Spieler kann mit Schwarz gegen den Computer spielen.

Alle Sonderzüge, Rochade, enPassant-Schlagen sowie die Bauernumwandlung sind möglich (Bauern werden grundsätzlich in eine Dame umgewandelt).

Unser Urteil: Ein sehr gutes, empfehlenswertes Programm, den meisten reinen Schachcomputern zumindest ebenbürtig, jedoch wesentlich preiswerter, als alles bisher angebotene.

Ein Spielprogramm für den VC-20 mit mindestens 8K Speichererweiterung.

Spielen können zwei Spieler gegeneinander oder ein Spieler allein gegen den Computer.

Ziel des Spieles ist es, das gegen den Uhrzeigersinn fahrende Auto mit den Tasten „“ und „.“ (innen,

außen) möglichst lange unfallfrei zu fahren. Wenn fast alle Punkte verbraucht sind, erhält man ein „Bonusauto“.

Beantwortet man die anfangs gestellte Frage nach der Spielerzahl mit „2“, so kann ein zweiter Spieler das im Uhrzeigersinn fahrende Crashauto gleichzei-

tig mit „C=“ (das ist die Commodore-Taste) und „Shift“ (innen, außen) steuern.

Vor dem Laden oder Eintippen muß

POKE44,32:POKE8182,0:
NEW eingegeben werden.

Georg Foltin

```

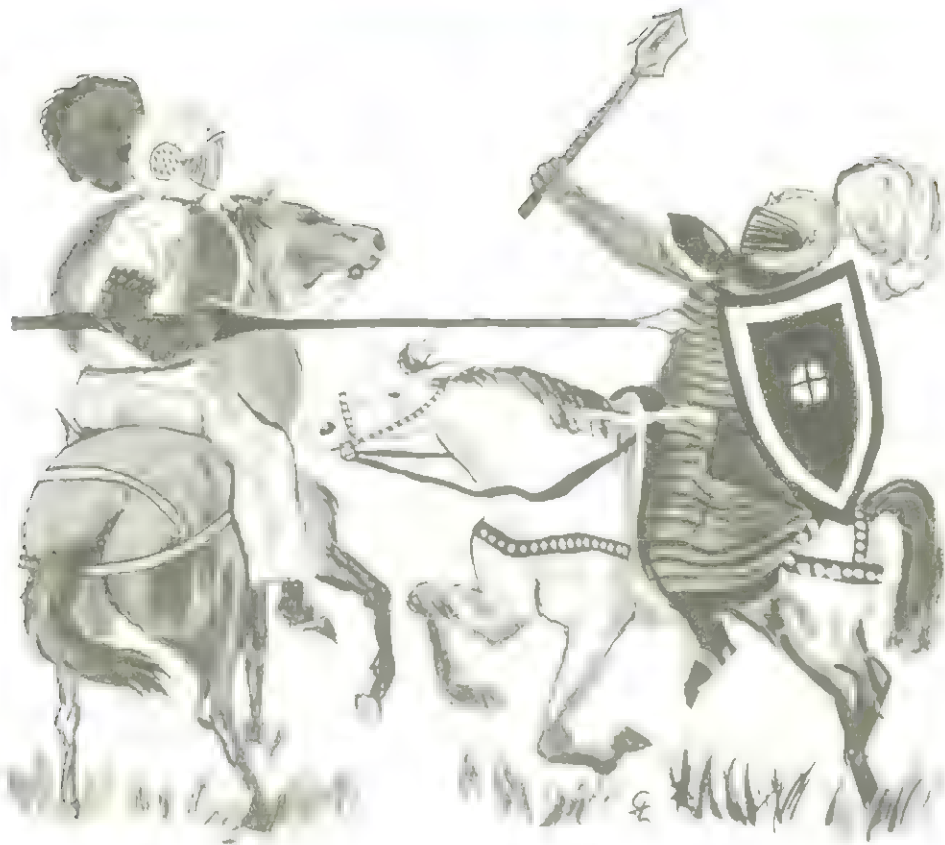
1 REM
2 REM CAR CRASH
3 REM
4 REM BY GEORG FOLTIN
5 REM
6 REM 1983
7 REM
8 REM LAENGE: 4.2K
9 REM
10 PRINT "CAR CRASH" CHR$(8); : POKE36879,25
20 FOR I=1 TO 21
25 POKE646,RND(1)*6+2:PRINT " "
30 NEXT
40 PRINT " "
41 PRINT TAB(5); " "
42 PRINT TAB(5); " " VC=20
43 PRINT TAB(5); " "
44 PRINT TAB(5); " CAR CRASH "
45 PRINT TAB(5); " "
50 PRINT " " : FOR I=1 TO 21
60 POKE646,RND(1)*6+2:PRINT " " : NEXT
70 GOSUB7000
80 PRINT " " PRINT " " (1) := 1 SPIELER"
85 PRINT " " (2) := 2 SPIELER
86 GETA$: IFA#<> 1"ANDA#<> 2" THEN 85
87 Y=VAL(A#+" ")
88 PRINT " " BITTE WARTEN"
100 GOSUB3000
110 REM BEWEGUNG MENSCH
120 POKEPM,32
130 AC=PEEK(CM+V(IM))
140 IFA#81 THEN GOSUB3000:ONS00TO110,200
150 IFA#32 THEN 200
160 IFA#PM+V(IM) THEN 4000
170 IM=(IM+1)AND3
200 POKEI,0
205 PM=PM+V(IM)
210 POKEPM,C(IM)
215 IFA#PM+V(IM) THEN 4000
220 POKEPC,0C
230 AC=PEEK(PC+V(IC))
240 IFA#81 OR AC#32 THEN 300
250 IFA#PC+V(IC) THEN 4000
260 IC=(IC-1)AND3
300 PC=PC+V(IC)

```


Starship VC-20


```
310 OC=PEEK(PC)*POKEPC*(OC)
315 IFP=PC*(OC) THEN GOTO 320
320 G=PEEK (197) : V=(V-1)*G+24
330 IFV<0 THEN GOTO 430
340 V=V*(10+INT(RND))
350 IFPEEK (PM+V)>32 THEN GOTO 430
355 POKET+1,190 : POKET,245
360 FORPM=32 TOPEEK*(LIM)
370 IFP+(V)*PC+MEMA=0
380 POKEPN+32 : PH=PH+V
390 POKEPN*(V)
400 POKET+1,0 : POKET,0
410 ONV=80 TO 451,460
425 IFPND*(V)>6 THEN GOTO 430
435 V=SGN(RND(1)-.5)
436 GOTO 447
440 V=PEEK (655) : V=(V-2)*V+1
445 IFV=0 THEN GOTO 450
470 V=V*(1+INT(RND))
480 IFPEEK (V+PC)>32 THEN GOTO 450
490 POKEPN,32 : POKEPN*(V)
500 IFPC=24 THEN MEMA=0
510 POKEPN+32 : PC=PC+24
520 POKEPN*(V)
550 GOTO 411
3000 S=2
5010 SC=SC+1 : POKET (S,INT(S)) : V
5020 PH=PH+V : MID$(T,1,SC)=V
3030 ZZ=ZZ+1
3040 IFZZ<140 THEN RETURN
3044 FORI=1 TO F : NEXT I : 220 POKET+V
3046 POKET (S,INT(S))
3050 ZZ=0 : S=1 : GOTO 411
3055 POKET,0
3060 FORI=655 TO 400
3070 IFI=INT (I/25)
3080 FET=I
4000 F1=INT (F) : POKET (F1,INT (F1))
4010 F2=INT (F) : POKET (F2,INT (F2))
4014 FORI=1 TO 100 : NEXT I
4015 POKET+1,190
4020 F1=INT (F)
4030 POKEPN*(F1) : POKET (F1,INT (F1))
4035 POKEPN*(F1) : POKET (F1,INT (F1))
4040 NEXT F
4045 POKET (I,INT (I))
4050 GOTO 411 : IFV=0 THEN GOTO 411 : GOTO 411 : GOTO 411
0
4050 POKET,198,0
4055 POKET (SC) : POKET (SC) : POKET (SC)
4070 POKET (I) : POKET (I)
4080 IFPEEK (PC+V)>32 THEN GOTO 411 : GOTO 411 : GOTO 411
4095 POKET (I) : POKET (I)
4100 POKET (I) : POKET (I)
4105 POKET (I) : POKET (I)
4110 GOTO 411
3000 FORI=1 TO 511 : POKET (I,INT (I)) : NEXT I
5000 POKET (I) : POKET (I) : POKET (I)
5005 POKET (I) : POKET (I) : POKET (I)
5010 POKET (I) : POKET (I) : POKET (I)
5040 POKET (I) : POKET (I) : POKET (I)
5050 POKET (I) : POKET (I) : POKET (I)
5055 POKET (I) : POKET (I) : POKET (I)
5060 POKET (I) : POKET (I) : POKET (I)
5070 POKET (I) : POKET (I) : POKET (I)
5080 POKET (I) : POKET (I) : POKET (I)
5090 POKET (I) : POKET (I) : POKET (I)
5100 POKET (I) : POKET (I) : POKET (I)
5110 POKET (I) : POKET (I) : POKET (I)
5120 POKET (I) : POKET (I) : POKET (I)
```

```
5130 PRINT : GOTO 411
5140 GOTO 411
5150 GOTO 411
5160 GOTO 411
5170 GOTO 411
5180 GOTO 411
5190 GOTO 411
5200 GOTO 411
5210 GOTO 411
5220 GOTO 411
5230 GOTO 411
5240 GOTO 411
5250 GOTO 411
5260 GOTO 411
5270 GOTO 411
5280 GOTO 411
5290 GOTO 411
5300 GOTO 411
5310 GOTO 411
7010 RESTORE
7020 GOTO 411
7030 GOTO 411
7040 GOTO 411
7050 GOTO 411
7060 GOTO 411
7070 GOTO 411
7080 GOTO 411
7100 NEXT J
7110 NEXT I
7120 GOTO 411
7130 GOTO 411
7140 GOTO 411
7150 GOTO 411
7160 GOTO 411
7170 GOTO 411
7180 GOTO 411
7190 GOTO 411
7200 GOTO 411
7210 GOTO 411
7220 GOTO 411
7230 GOTO 411
7240 GOTO 411
7250 GOTO 411
7260 GOTO 411
7270 GOTO 411
7280 GOTO 411
7290 GOTO 411
7300 GOTO 411
7310 GOTO 411
7320 GOTO 411
7330 GOTO 411
7340 GOTO 411
7350 GOTO 411
7360 GOTO 411
7370 GOTO 411
7380 GOTO 411
7390 GOTO 411
7400 GOTO 411
7410 GOTO 411
7420 GOTO 411
7430 GOTO 411
7440 GOTO 411
7450 GOTO 411
7460 GOTO 411
7470 GOTO 411
7480 GOTO 411
7490 GOTO 411
7500 GOTO 411
7510 GOTO 411
7520 GOTO 411
7530 GOTO 411
7540 GOTO 411
7550 GOTO 411
7560 GOTO 411
7570 GOTO 411
7580 GOTO 411
7590 GOTO 411
7600 GOTO 411
7610 GOTO 411
7620 GOTO 411
7630 GOTO 411
7640 GOTO 411
7650 GOTO 411
7660 GOTO 411
7670 GOTO 411
7680 GOTO 411
7690 GOTO 411
7700 GOTO 411
7710 GOTO 411
7720 GOTO 411
7730 GOTO 411
7740 GOTO 411
7750 GOTO 411
7760 GOTO 411
7770 GOTO 411
7780 GOTO 411
7790 GOTO 411
7800 GOTO 411
7810 GOTO 411
7820 GOTO 411
7830 GOTO 411
7840 GOTO 411
7850 GOTO 411
7860 GOTO 411
7870 GOTO 411
7880 GOTO 411
7890 GOTO 411
7900 GOTO 411
7910 GOTO 411
7920 GOTO 411
7930 GOTO 411
7940 GOTO 411
7950 GOTO 411
7960 GOTO 411
7970 GOTO 411
7980 GOTO 411
7990 GOTO 411
8000 GOTO 411
8010 GOTO 411
8020 GOTO 411
8030 GOTO 411
8040 GOTO 411
8050 GOTO 411
8060 GOTO 411
8070 GOTO 411
8080 GOTO 411
8090 GOTO 411
8100 GOTO 411
8110 GOTO 411
8120 GOTO 411
8130 GOTO 411
8140 GOTO 411
8150 GOTO 411
8160 GOTO 411
8170 GOTO 411
8180 GOTO 411
8190 GOTO 411
8200 GOTO 411
8210 GOTO 411
8220 GOTO 411
8230 GOTO 411
8240 GOTO 411
8250 GOTO 411
8260 GOTO 411
8270 GOTO 411
8280 GOTO 411
8290 GOTO 411
8300 GOTO 411
8310 GOTO 411
8320 GOTO 411
8330 GOTO 411
8340 GOTO 411
8350 GOTO 411
8360 GOTO 411
8370 GOTO 411
8380 GOTO 411
8390 GOTO 411
8400 GOTO 411
8410 GOTO 411
8420 GOTO 411
8430 GOTO 411
8440 GOTO 411
8450 GOTO 411
8460 GOTO 411
8470 GOTO 411
8480 GOTO 411
8490 GOTO 411
8500 GOTO 411
8510 GOTO 411
8520 GOTO 411
8530 GOTO 411
8540 GOTO 411
8550 GOTO 411
8560 GOTO 411
8570 GOTO 411
8580 GOTO 411
8590 GOTO 411
8600 GOTO 411
8610 GOTO 411
8620 GOTO 411
8630 GOTO 411
8640 GOTO 411
8650 GOTO 411
8660 GOTO 411
8670 GOTO 411
8680 GOTO 411
8690 GOTO 411
8700 GOTO 411
8710 GOTO 411
8720 GOTO 411
8730 GOTO 411
8740 GOTO 411
8750 GOTO 411
8760 GOTO 411
8770 GOTO 411
8780 GOTO 411
8790 GOTO 411
8800 GOTO 411
8810 GOTO 411
8820 GOTO 411
8830 GOTO 411
8840 GOTO 411
8850 GOTO 411
8860 GOTO 411
8870 GOTO 411
8880 GOTO 411
8890 GOTO 411
8900 GOTO 411
8910 GOTO 411
8920 GOTO 411
8930 GOTO 411
8940 GOTO 411
8950 GOTO 411
8960 GOTO 411
8970 GOTO 411
8980 GOTO 411
8990 GOTO 411
9000 GOTO 411
9010 GOTO 411
9020 GOTO 411
9030 GOTO 411
9040 GOTO 411
9050 GOTO 411
9060 GOTO 411
9070 GOTO 411
9080 GOTO 411
9090 GOTO 411
9100 GOTO 411
9110 GOTO 411
9120 GOTO 411
9130 GOTO 411
9140 GOTO 411
9150 GOTO 411
9160 GOTO 411
9170 GOTO 411
9180 GOTO 411
9190 GOTO 411
9200 GOTO 411
9210 GOTO 411
9220 GOTO 411
9230 GOTO 411
9240 GOTO 411
9250 GOTO 411
9260 GOTO 411
9270 GOTO 411
9280 GOTO 411
9290 GOTO 411
9300 GOTO 411
9310 GOTO 411
9320 GOTO 411
9330 GOTO 411
9340 GOTO 411
9350 GOTO 411
9360 GOTO 411
9370 GOTO 411
9380 GOTO 411
9390 GOTO 411
9400 GOTO 411
9410 GOTO 411
9420 GOTO 411
9430 GOTO 411
9440 GOTO 411
9450 GOTO 411
9460 GOTO 411
9470 GOTO 411
9480 GOTO 411
9490 GOTO 411
9500 GOTO 411
9510 GOTO 411
9520 GOTO 411
9530 GOTO 411
9540 GOTO 411
9550 GOTO 411
9560 GOTO 411
9570 GOTO 411
9580 GOTO 411
9590 GOTO 411
9600 GOTO 411
9610 GOTO 411
9620 GOTO 411
9630 GOTO 411
9640 GOTO 411
9650 GOTO 411
9660 GOTO 411
9670 GOTO 411
9680 GOTO 411
9690 GOTO 411
9700 GOTO 411
9710 GOTO 411
9720 GOTO 411
9730 GOTO 411
9740 GOTO 411
9750 GOTO 411
9760 GOTO 411
9770 GOTO 411
9780 GOTO 411
9790 GOTO 411
9800 GOTO 411
9810 GOTO 411
9820 GOTO 411
9830 GOTO 411
9840 GOTO 411
9850 GOTO 411
9860 GOTO 411
9870 GOTO 411
9880 GOTO 411
9890 GOTO 411
9900 GOTO 411
9910 GOTO 411
9920 GOTO 411
9930 GOTO 411
9940 GOTO 411
9950 GOTO 411
9960 GOTO 411
9970 GOTO 411
9980 GOTO 411
9990 GOTO 411
```



Ritter

Ihre Aufgabe ist es eine Burg einzunehmen. Sie können sich mit den Tasten:

- I ← links
- E → rechts
- P  Schild
- L  Schuß bewegen.

Sinn des Spieles ist es, die Kette der Zugbrücke durchzuschießen, indem Sie ganz nach vorne zum Graben gehen und dann auf „Teufel komm raus“ losfeuern.

Es ist natürlich klar, daß sich die Burgbewohner so etwas nicht gefallen lassen. Sie werden sich mit Bogenschützen und Wunderwolken verteidigen.

Wenn ein Pfeil auf Sie zufliegt, müssen Sie sich ducken „P“.

Bei der Wunderwolke schützen Sie sich entweder mit dem Schutzschild „P“ vor dem Pechregen, aber besser ist es ihm auszuweichen.

Sie sollen die Kettenbefestigung 16 mal treffen. Wenn Sie sich ducken oder das Schutzschild benutzen, wird Ihnen jeweils 1-20 Treffer abgezogen.

Sie fangen mit 0 Treffern an.

Sie sollen 16 Treffer erreichen und können maximal -6 Treffer durch ducken oder Schutzschildbetätigung bekommen. Jedemal wenn Sie sich ducken wird nämlich die Kette erneuert bzw. repariert.

Zum Programm:

Als erstes geben Sie ein:
1 Rem... (19. Punkte. ein).
N/L

Danach schreiben Sie
Poke 16510,0 N/L
und dann geben Sie das

Listing wie gewohnt der Reihe nach ein.

Bemerkungen:

Bevor Sie Ihren ersten Probelauf machen, save Sie lieber das Programm, denn wenn in den ersten 7 Zeilen ein Fehler ist, kann dann das Programm gelöscht werden, wenn das MC-Programm in Zeile 560 und 4010 abgerufen wird.

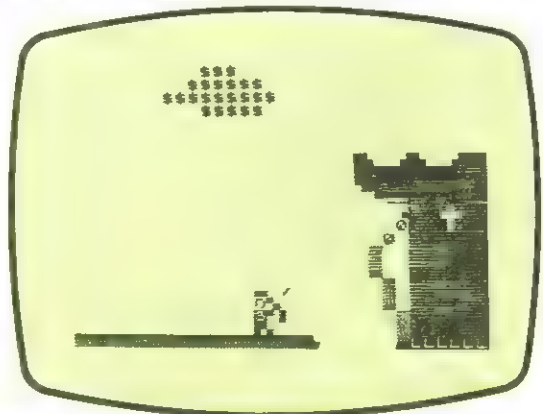
In Zeile 510 wird man in den Gefahrenabschnitt ge-

bracht. Durch verändern des Wertes hinter dem „RND“ beeinflußt man die Gefahren.

In Zeile 570 wird bestimmt, ob man dem Bogenschützen oder der Wolke ausgesetzt wird.

In der Zeile 420 werden die Inputzeilen zu den Ausdruckzeilen dazugegeben (Bildschirm) (24-Zeilen-Bildschirm)

Stefan Benk



Die ZX-Seiten

```

0 REM ERAND: 477$4 NEXT TAN
1 PRINT AT 0,0;" EROBERN SI
E EINE BURG... GE
STEUERT WIRD MIT:
1.....L
E.....RE
CHTS.....3C
HUSS.....5C
HUTZ",TAB 3;"UIEL GLUECK"
2 RAND
3 LET A$="A20004E081600232E7E
E0877019F32D00023F9C"
4 FOR A=16514 TO 16532
5 POKE A,CODE A$+16*CODE A$(2
)-476
6 LET A$=A$(3 TO )
7 NEXT A
8 FOR A=1 TO 50
9 NEXT A
10 LET A$=""
11 LET A$=A$+" "
20 LET A$=A$+" "
30 LET A$=A$+" "
40 LET A$=A$+" "
50 LET A$=A$+" "
60 LET A$=A$+" "
70 LET A$=A$+" "
80 LET A$=A$+" "
90 LET A$=A$+" "
100 LET A$=A$+" "
110 LET A$=A$+" "
120 LET A$=A$+" "
130 LET A$=A$+" "
140 LET A$=A$+" "
150 LET A$=A$+" "
160 LET A$=A$+" "
170 LET A$=A$+" "
180 REM " 0 "
190 REM " 1 "
200 REM " 2 "
210 REM " 3 "
220 REM " 4 "
230 REM " 5 "
240 REM " 6 "
250 REM " 7 "
260 REM " 8 "
270 REM " 9 "
280 REM " "
290 REM " 0 "
300 REM " 1 "
310 REM " 2 "
320 REM " 3 "
330 REM " 4 "
340 REM " 5 "
350 REM " 6 "
360 REM " 7 "
370 REM " 8 "
380 REM " 9 "
390 REM " 0 "
400 REM " 1 "
410 REM " 2 "
420 POKE 16418,0
425 CLS
430 PRINT AT 7,0;A$;AT 17,3;"
/TAB 3;" 0";TAB 3;" 1";
TAB 3;" 2";
440 LET A=4
450 LET L=0

```

```

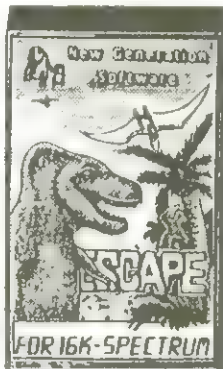
460 LET S=0
470 LET D=0
480 LET B=1
490 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
500 PRINT AT 17,A;" / ";TAB A
;" 0";TAB A;" 1";TAB A;" 2";
510 IF AND<.15 THEN GOSUB 550
520 LET D=INKEY$="L"
530 IF D OR S THEN GOSUB 1080
540 GOTO 490
550 REM "ACHTUNG"
560 PRINT TAB USA 16514;AT 22,2
;"ACHTUNG...";TAB USA 16514
570 IF AND>.5 THEN GOTO 730
580 REM "GOSSENSCHUTZE"
590 PRINT AT 16,24;" ";TAB
24;" <- 0 ";TAB 22;" ";T
AB 22;" "
600 FOR C=31 TO 0 STEP -2
610 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
620 IF B THEN PRINT AT 17,A;" /
";TAB A;" 0";TAB A;" 1";
TAB A;" 2";
630 IF NOT B THEN PRINT AT 17,A
;" ";TAB A;" ";TAB A;" 0
";TAB A-1;" ";TAB
640 LET B=INKEY$<>"P"
650 IF NOT B THEN LET L=L-(L>-6
)
655 LET D=INKEY$="L"
660 IF D OR S THEN GOSUB 1080
670 IF C<25 THEN PRINT AT 17,C;
"<- "
680 IF ABS (A-C+1) < 2 AND B THEN
GOTO 2000
690 NEXT C
700 PRINT AT 17,0;" "
710 PRINT AT 22,2;" ";
AT 16,24;" ";TAB 24;" ";
";TAB 22;" ";TAB 22;" "
720 RETURN
730 REM "BECH"
740 FOR C=22 TO A-2 AND A>3 STE
P -1
750 PRINT AT 0,C+3;"$$$ ";TAB C
+2;" $$$$ ";TAB C;" $$$$$$$$ ";
TAB C+3;" $$$$ "
760 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
770 LET B=INKEY$<>"P"
775 IF NOT B THEN LET L=L-(L>-6
)
780 IF B THEN PRINT AT 17,A;" /
";TAB A;" 0";TAB A;" 1";
TAB A;" 2";
790 IF NOT B THEN PRINT AT 17,A
-1;" ";TAB A-1;" ";TAB
A;" ";TAB A;" ";
795 LET D=INKEY$="L"
800 IF D OR S THEN GOSUB 1080
810 NEXT C
820 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
830 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
840 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
850 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
860 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
870 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
880 PRINT AT 0,C+3;" $ ";TAB
C+2;" $ ";TAB C+1;" $ "

```

ALLES
FÜR

ZX81 UND SPECTRUM

VON
F+K



TUNNEL

Reaktionsspiel mit superschneller 3D-Farbgraphik. Für SPECTRUM 16K/48K. Was liegt im TUNNEL verborgen? DM 29,-

ESCAPE

3D Adventure. Bis zu 5 Monster sind hinter Ihnen her, während Sie den Ausgang des Irrgartens suchen. Tolle Graphik! Für SPECTRUM 16K/48K.

nur DM 27,-



3D DEFENDER für ZX81/16K BestNr 152
Vergessen Sie alles, was Sie bisher an Weltraumspielen gesehen haben. Bei 3D DEFENDER schauen Sie aus Ihrem Cockpit und sehen -dank der superschnellen Graphik- alles wie bei einem Flugsimulator vor sich. Und das in dreidimensionaler Darstellung. Mit 8 Flugrichtungen, Radar, UFOs, Plasmaangriffen, ... Das müßten Sie sehen!
"Another 3D winner" schreibt SINCLAIR USER.

3D MONSTER MAZE für ZX81/16K BestNr 151
Unglaubliche Graphik! COMPUTER & VIDEO GAMES schreibt: "3D MONSTER MAZE is the best game I have seen for the ZX81". Finden Sie Ihren Weg heraus aus dem Irrgarten, während REX Ihnen nachjagt? Wände, Gänge und REX selbst sind dreidimensional dargestellt, sowas ist noch nie dagewesen. Nur bei uns, nur DM 29,-

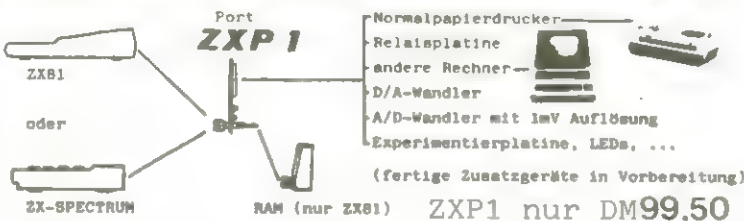


"If I had to choose just one programme to impress an audience with the capabilities of the ZX81, then 3D MONSTER MAZE would be the one without doubt" schreibt ZX COMPUTING.

Port ZXP1 für ZX81, ZX-SPECTRUM BestNr 400

Technische Daten: Eingabe 8 Bit, Ausgabe 8 Bit, zusätzlich 5 STROBEs. Hochstromausgänge (low-24mA), Schmitt-Trigger-Eingänge. Anschlüsse steckbar. Fertiger mit ohne Gehäuse mit ausführlicher Anleitung.

Besonderheiten: für den Druckeranschluß (GPIOA) vorbereitet (CENTRONICS)



(fertige Zusatzgeräte in Vorbereitung)

RAM (nur ZX81) ZXP1 nur DM99,50

Zusätzlich kann auch der SINCLAIR-Drucker angeschlossen werden.

DBMS 81

Programm zur Erzeugung beliebiger Dateien/Kartellen. Unterstützt Sie nach der Erzeugung bei der Verwaltung Ihrer Daten, und ermöglicht durch eine spezielle Speicherverwaltung eine bessere Ausnutzung des RAMs. Z.B. werden über 15500 Zeichen im 16K-RAM verwaltet, im 64K-RAM sind es über 60000 IIII!

Ähnliche Systeme verwalten sonst nur ca. 10000 Zeichen im 16K-RAM. Weitere Details in unserer Info. DBMS81 für ZX81/ mindestens 16K äußerst preiswert nur DM 45,-

INKA - TEMPEL



Nur bei uns: das erste SPECTRUM-Adventure in deutsch und in MC. Sie bannen sich Ihren Weg durch 53 Räume, um die 7 Schätze des Tempels einzusammeln. Aber das ist schwerer, als es Ihnen jetzt vorkommt! Nur für 48K-SPECTRUM.

nur DM
29,-

Wir führen auch **HARDWARE**
S. CHIP 4/83 S. 179

Der Knüller APPEND

APPEND ermöglicht das Zusammenladen von verschiedenen BASIC-Programmen. Außerdem ist noch eine RENUMBER-Funktion eingebaut, die Ihre Programme oder Teile davon neu durchnummerieren kann (auch GOTO, GOSUB und RUN).

Unser APPEND ist das einzige derartige Programm, mit dem man z.B. auch 2 Programme à 5 KByte zusammenladen kann. Andere Programme gleichen Namens können das nicht!!

APPEND ist nicht billig, denn Qualität hat ihren Preis, aber es lohnt sich!

APPEND für ZX81/16K DM 45,-

Alle diese Produkte kommen aus dem Hause Wolfgang Fleischhauer und Nils Körber GbR.

Wollen Sie mehr wissen? Wir schicken Ihnen gerne eine gegen Voreinsendung eines selbstadressierten Freiumschlages DIN C6.

Schicken Sie diesen an:

» F+K-WARE, Abt. H
Rebenacker 1a, 2000 Hamburg 54 «

SPECTRUM-Benutzer fordern bitte 'Liste S' an.

für ZX81/16K:
Spiele-Klassiker

- ASTEROIDEN DM 21,-
- INVADERS DM 21,-
- BREAKOUT* DM 12,95
- FUCKMAN DM 27,-
- CENTIPEDE DM 29,-

*auch für ZX81/1K

ORDER

Bestellungen bitte auch an die vorgenannte Adresse richten. Versandpauschale: DM 6,-. Alle genannten Preise enthalten die ges. MwSt und sind gültig bis 30.9.83.

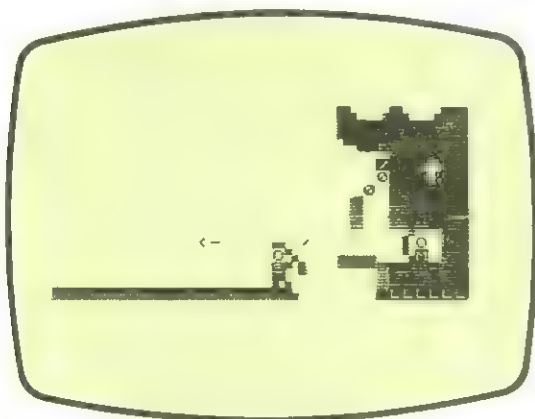
Händleranfragen erwünscht!!

Die ZX-Seiten

```

";TAB C+3;"$
881 PRINT AT 0,C+3;" ";TAB C+2
";TAB C+1;"$";T
AB C+3;"$
882 PRINT AT 1,C+2;" ";TAB
C+1;"$";TAB C+3;"$";TAB
883 PRINT AT 1,C+2;" ";TAB
C+1;" ";TAB C+3;"$";TAB
884 PRINT AT 3,C+3;" ";TAB C
+3;"$";TAB C
885 PRINT AT 4,C+3;" ";TAB C
+3;"$";TAB C
890 FOR Z=3 TO 18 STEP 2
900 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
910 LET D=INKEY$="L"
920 IF D OR S THEN GOSUB 1080
930 LET B=INKEY$<>"P"
940 IF B THEN PRINT AT 17,A;"
/" ;TAB A;" 0";TAB A;"$";TAB
TAB A;"$";TAB A;"$";TAB A;"$";
950 IF NOT B THEN PRINT AT 17,A
-1;"$";TAB A-1;" 0";TAB
A;"$";TAB A;"$";TAB A;"$";TAB
960 IF NOT B THEN LET L=L-(L)-6
)

```



```

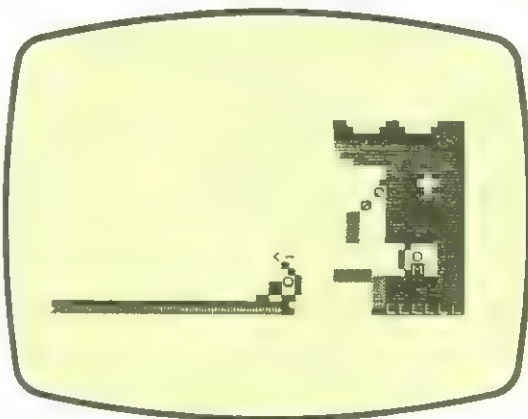
970 PRINT AT Z,C+4;"$";AT Z-2,C
+4,
980 NEXT Z
990 PRINT AT Z-2,C+4,
995 PRINT AT 22,0,
1000 IF ABS (A-C-3) > 1.6 OR NOT B
THEN RETURN
1005 LET L=RND**RND**RND
1010 PRINT AT 17,A-1;"$";TA
B A-1;"$";TAB A-1;"$";TAB
TAB A-1;"$";TAB A;"$";TAB
1015 LET L=RND**RND**RND
1020 PRINT AT 17,A;"$";TAB A-
1;"$";TAB A-1;"$";TAB
A-1;"$";TAB A-1;"$";TAB
B A;"$";TAB
1025 LET L=RND**RND**RND
1026 PRINT AT 13,A;"$";TAB A-
1;"$";TAB A-1;"$";TAB
A-1;"$";TAB A-1;"$";TAB
B A;"$";TAB
1027 LET L=RND**RND**RND
1030 PRINT AT 0,0;"DER ""RECH-AB
SCHLAGEN"" HAT SIE ""ER
SCHLAGEN""
1040 GOTO 2050
1080 IF S THEN GOTO 1140
1090 LET H=18
1100 FOR S=A+3 TO A+9
1110 LET H=H-1
1120 PRINT AT H+1,S-1,AT H,S;"/"
.
1130 RETURN
1140 NEXT S
1150 PRINT AT H,S-1,
1160 LET L=L+(S=26)
1170 IF S=26 THEN PRINT AT 11,25
;CHR$(L+156)

```

```

1180 LET S=0
1190 IF L=16 THEN GOTO 4000
1200 LET D=0
1210 RETURN
2000 PRINT AT 17,A;" / ";TAB A
;" 0";TAB A;"$";TAB A;"$";
2005 LET L=RND**RND**RND
2010 PRINT AT 17,A;" / ";TAB A
;" 0";TAB A;"$";TAB A;"$";
2015 LET L=RND**RND**RND
2020 PRINT AT 17,A;" / ";TAB A
;" 0";TAB A;"$";TAB A;"$";
2025 LET L=RND**RND**RND
2030 PRINT AT 17,A;" / ";TAB A
;"$";TAB A;"$";TAB A-1;"0
$";
2035 LET L=RND**RND**RND
2040 PRINT AT 0,0;"VOM BOGENSCHU
ETZEN ERWISCHT"
2050 PRINT AT 10,0;"NOCHMAL?"
2060 RUN (2060 AND INKEY$<>"J")
4000 FOR Z=0 TO 30
4010 LET L=USR 16514+USR 16514
4020 NEXT Z
4030 PRINT AT 22,1;"GENONNEN..."
;TAB C;"$";TAB C;"$";
4040 FOR Z=14 TO 20
4050 PRINT AT Z,23;" ";
4060 NEXT Z
4070 PRINT AT 20,19;" ";
4080 PRINT AT 21,18;" ";
4090 FOR A=A TO 25
4100 PRINT AT 17,A;" / ";TAB A
;" 0";TAB A;"$";TAB A;"$";
4110 NEXT A
4115 PRINT AT 1,0;"HERZLICHEN G
LIEDERLEBEN HERR RITTER"
4120 PRINT AT 7,0;A$( TO LEN A$-
64)
4130 PRINT AT 10,0;"NOCHMAL?"
4140 RUN (4150 AND INKEY$<>"J")
4150 PRINT AT 6,22-6;"$";0

```



```

";TAB 22;"$";
4155 LET L=RND**RND
4160 PRINT AT 6,22-6;" 0
";TAB 22;"$";
4170 PRINT AT 6,22-6;" 0
";TAB 22;"$";
4180 PRINT AT 6,22-6;" 0
";TAB 22;"$";
4190 PRINT AT 6,22-6;" 0
";TAB 22;"$";
4200 PRINT AT 6,22-6;" 0
";TAB 22;"$";
4210 GOTO 4140
9940 STOP
9950 SAVE "RITTER"
9960 RUN

```

Missile ZX-Command

In Mußestunden habe ich versucht, das bekannte Spielhöllenspiel "Missile Command" so gut es mir möglich war, auf dem ZX-81 zu gestalten.

Zur Eingabe:

Wenig angetan von der fehlerträchtigen Eingabeart eines Hex-Dumps in CS, habe ich durch Ausdruck der dezimalen Äquivalente und einer entsprechenden Eingaberoutine (Zeilen 8000-8004) den Versuch gemacht, sowohl die Eingabe schneller, als auch die Fehlerrate (z.B. Verwechslung von B und 8) geringer zu machen. Das hat obendrein den Vorteil, daß keine Initialisierung notwendig ist.

Nach einmaliger Eingabe der dezimalen Liste (von links nach rechts!) kann die Eingabe-Routine gelöscht werden, da die REM-Zeile ja mitgesaved wird.

Beim ersten Start des Programms mit RUN muß zuerst die dezimale Datentabelle eingetippt werden. Bevor man dann das Programm ein für alle mal absaven kann, müssen die Zeilen 8000 - 8004 gelöscht werden. Jeder weitere Start erfolgt dann mit RUN.

Zum Spiel:

Die Aufgabe des Spielers ist es, möglichst lange die 6 Städte vor den Laserstrahlen zu schützen. Das kann er dadurch erreichen, daß er den grauen Zielbalken unter einen der Strahlen bewegt und den Feuerknopf betätigt.

An der Stelle des Zielbalkens wird dann eine Explosion entstehen, die den Strahl zerstört. Bis zu 3 Explosionen können gleichzeitig auf dem Bildschirm sein.

Es steht nur eine begrenzte Anzahl an Schüssen zur Verfügung. Sie werden unten als Pluspunkte dargestellt. Zusätzlich geht Energie verloren, wenn ein Strahl die Basisstadt in der Mitte trifft. Nach 50 Strahlen erfolgt eine Zwischenwertung. Das Spiel wird dann automatisch schneller fortgesetzt.

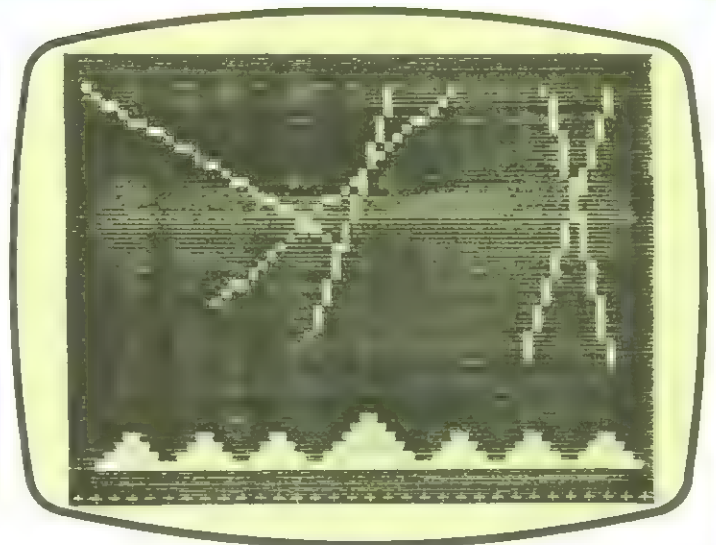
In der Version, die durch das Listing angegeben wird, haben die Tasten folgende Bedeutungen (Ziffern links).

6	Zielbalken runter	D
7	Zielbalken rauf	O
8	Zielbalken links	W
9	Zielbalken rechts	K
0	Feuer	O

Die Buchstaben (rechts) habe ich so vereinbart, weil ich parallel zu diesen Tasten einen ATARI-Joystick angeschlossen habe. Für ZX-Besitzer, die das nicht haben, wird durch folgende Befehle eine wesentlich besser zu greifende Belegung erreicht:

```
POKE 16561,117
POKE 16556,85
POKE 16566,109
POKE 16571,77
```

Martin Zwerschke



```

1 REM DAS ZEICHEN (37 ZEI
LEN
2 REM MISSILE COMMAND BY
MARTIN ZWERSCHKE 3/83

10 REM
20 GOTO 8000
1000 PRINT AT 0,9; "
";TAB 9;"";TAB 21;"";TAB 9;"
GAME OVER";TAB 9;"";TAB 21;"
";TAB 9;"
1010 IF INKEY$="" THEN GOTO 1010
1020 RUN
2130 LET S=26033
2140 LET A$="1112112313122332123
1221144"
2200 FOR I=1 TO LEN A$
2220 LET S=S+1
2240 POKE S,VAL A$(I)
2260 NEXT I
2270 POKE S,0
2280 LET A=PEEK 26000
2290 LET B=PEEK 26001
2300 POKE 26022,A
2320 POKE 26023,B
2340 POKE 26025,A
2360 POKE 26027,B
2380 POKE 26030,A
2400 POKE 26031,B
3000 LET L=USR 16514
3100 LET L=USR 17533
3205 PRINT AT 0,0
3210 FOR I=0 TO 15
3220 PRINT "
3230 NEXT I
3240 IF PEEK 27030>3 THEN POKE 2
7030,PEEK 27030-3
3300 LET PUNKTE=PEEK 27024+256*P
EEK 27025
3310 PRINT AT 22,15:PUNKTE
3320 LET DF=PEEK 16396+256*PEEK
16397
3330 LET STAEDETE=0
3340 LET B$=""
3350 FOR I=DF+693 TO DF+792
3360 IF PEEK I<>118 THEN LET B$=
B$+CHR$ PEEK I
3370 NEXT I
3380 LET I=4
3382 GOSUB 4E3
3384 LET I=8
3386 GOSUB 4E3
3388 LET I=12
3390 GOSUB 4E3
3392 LET I=22
3394 GOSUB 4E3
3396 LET I=26
3398 GOSUB 4E3
3400 LET I=30

```


Die ZX-Seiten

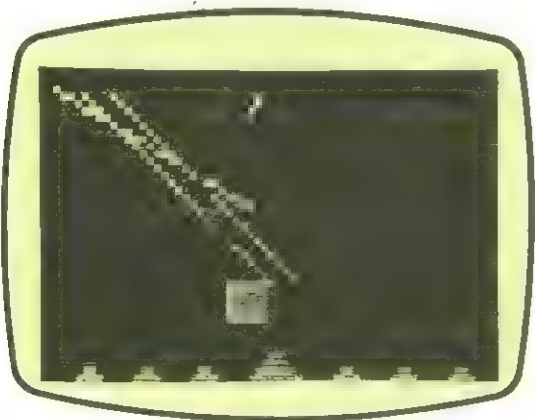
```

3402 GOSUB 4E3
3420 FOR I=65 TO 96
3430 IF B$(I)="■" THEN GOTO 5000
3435 GOSUB 3600
3440 NEXT I
3450 GOTO 5000
3600 PRINT AT 23,I-65;"■"
3610 LET PUNKTE=PUNKTE+1
3620 PRINT AT 22,15;PUNKTE
3630 RETURN
4000 IF B$(I)<>" " THEN RETURN
4010 PRINT AT 21,I-3;"■";AT
22,I-3;"■"
4020 LET PUNKTE=PUNKTE+10
4030 LET STAEDETE=STAEDETE+1
4032 FOR I=1 TO 14
4034 NEXT I
4036 PRINT AT 22,15;PUNKTE
4040 RETURN
5000 LET ABF=1
5001 IF STAEDETE=0 THEN GOTO 1000
5002 PRINT AT 21,0;B$
5004 POKE 27025,INT (PUNKTE/256)
5006 POKE 27024,PUNKTE-256*PEEK
27025
5008 PRINT AT 22,15;PUNKTE
5010 GOTO 9000
8000 FOR I=16514 TO 17689
8001 INPUT A
8002 POKE I,A
8003 PRINT I,PEEK I
8004 NEXT I
8008 POKE 16418,2
8009 CLS
8010 POKE 27024,0
8011 POKE 27025,0
8021 PRINT "STRAHLEN? (4/5)"
8022 INPUT STR
8023 POKE 27034,STR=5
8024 PRINT "GESCHWINDIGKEIT ?
(1-10)"
8025 INPUT G
8026 IF G<1 OR G>10 THEN GOTO 80
24
8027 POKE 27030,102-10*G
8030 CLS
8999 GOTO 9998
9000 FOR I=27600 TO 27625
9005 POKE I,0
9020 NEXT I
9030 POKE 26012,1
9035 POKE 16418,0
9039 IF ABF THEN GOTO 2000
9040 PRINT AT 21,0;"■";AT
22,0;"■";AT 23,0;"■";AT
9042 PRINT AT 22,0;"■";AT
23,0;"■";AT
9044 PRINT "■";AT 21,0;"■";AT
22,0;"■";AT
9046 PRINT AT 20,0;"■";AT
21,0;"■";AT
9997 GOTO 2000
9998 LET ABF=0
9999 GOTO 9000

```

DATEN :

03	200	101	04	160	101	04	160	101	04
101	34	176	101	101	00	00	00	00	00
101	42	10	04	04	00	00	00	00	00
04	140	101	1	04	04	04	04	04	04
244	101	00	004	10004	10000	10000	10000	10000	10000
204	200	04	0004	0000	0000	0000	0000	0000	0000
200	100	004	000	000	000	000	000	000	000
05	40	100	000	000	000	000	000	000	000
05	105	00	000	000	000	000	000	000	000
126	254	118	104	00	00	00	00	00	00
03	05	005	14	00	00	00	00	00	00
175	237	00	237	01	148	101	101	101	101
237	02	230	041	04	04	04	04	04	04
03	0	0	007	001	140	101	101	101	101
237	02	000	10	00	100	100	100	100	100
05	105	00	00	04	104	104	104	104	104
144	101	126	254	100	40	00	00	00	00
128	201	08	150	101	110	001	001	001	001
05	1	000	00	00	00	00	00	00	00
101	175	237	02	200	00	00	00	00	00
237	06	005	04	144	101	101	101	101	101
150	101	00	100	04	04	04	04	04	04
40	07	04	000	00	04	04	04	04	04
40	10	04	1	140	00	00	00	00	00
237	177	224	40	04	100	00	00	00	00
247	42	166	101	00	00	00	00	00	00
10	167	40	1	00	00	00	00	00	00
101	205	140	65	40	100	101	101	101	101
05	170	101	10	167	40	00	00	00	00
237	67	172	101	200	140	00	00	00	00
174	101	237	05	170	101	10	10	10	10
40	1	0	237	07	170	101	10	10	10
140	65	201	000	00	04	40	40	40	40
254	0	40	00	00	04	40	40	40	40
00	0	114	1	04	00	170	114	114	114
66	205	0	06	40	00	00	00	00	00
114	05	205	0	00	00	40	1	1	1
1	31	0	0	00	00	00	00	00	00
1	114	05	05	00	00	00	00	00	00
1	114	0	205	00	00	00	40	1	1
114	05	114	05	00	00	00	00	00	00
114	201	04	00	40	00	00	00	00	00
40	0	04	00	00	00	00	00	00	00
237	55	04	00	00	00	00	00	00	00
0	66	40	00	04	00	00	00	00	00
0	201	54	0	00	00	101	101	101	101
151	126	254	110	100	00	00	00	00	00
201	58	146	100	107	00	00	00	00	00
61	50	146	105	40	100	101	101	101	101
155	101	126	167	40	00	40	10	10	10
101	1	170	101	126	107	40	10	10	10
42	176	101	1	174	101	101	101	101	101
40	3	219	255	201	40	144	101	144	144
125	2	0	124	0	00	00	00	00	00
178	124	2	0	100	00	00	00	00	00
00	66	50	100	107	00	00	00	00	00
105	237	01	120	105	100	00	40	00	00
122	148	71	40	4	00	100	100	100	100
199	50	125	105	50	107	100	100	100	100
167	40	4	20	100	00	00	00	00	00
124	105	50	120	100	100	00	00	00	00
40	2	237	00	05	100	00	00	00	00
40	2	237	00	07	147	00	00	00	00
175	00	120	100	100	00	00	00	00	00
4	175	00	127	100	00	00	00	00	00
58	119	105	71	4	00	100	100	100	100
147	50	10	70	00	100	100	100	100	100
111	58	125	105	100	100	100	100	100	100
58	106	105	103	111	00	100	100	100	100
102	100	10	00	00	00	00	00	00	00
1	200	40	00	00	00	40	00	00	00
41	40	1	7	00	00	40	40	40	40
64	17	223	255	200	10	00	00	00	00
125	05	7	254	100	40	00	00	00	00
40	2	000	140	00	41	00	00	00	00
155	101	200	70	00	00	00	10	00	00
208	47	160	200	40	00	00	00	00	00
238	143	110	100	00	100	100	100	100	100
00	3	00	00	00	100	00	00	00	00
00	62	0	00	00	1	00	00	00	00
200	107	17	110	100	1	00	00	00	00
237	176	04	2	04	044	04	140	140	140



64' VideoChips

Mondlandung VC-64

Ein Raumschiff soll auf der Oberfläche des Mondes weich gelandet werden. Bei zu hartem Aufsetzen, bei zu starker seitlicher Abdrift oder beim Berühren der Felsen explodiert es. Wenn es zu weit aus dem Bildschirm verschwindet, hat es sich verirrt, was ebenfalls Minuspunkte bringt. Die Punktzahl errechnet sich nach der Schwierigkeit des Landeplatzes, nach der Entfernung des Landeplatzes vom Ausgangspunkt, nach dem noch vorhandenen Treibstoffvorrat und nach der Landegeschwindigkeit. Bei mehreren Spielen erscheint für jeden Teilnehmer im selben Durchgang die gleiche Landschaft, die vom Zufallsgenerator gezeichnet wird.

Die Steuerung erfolgt wahlweise mittels Joystick (in Control-port 1) rechts und links mit dem Steuerknüppel, die Bremsraketen werden mit der Feuertaste gezündet, oder über die Tastatur. Nach links mit ←, nach rechts mit →. Gebremst wird mit der Space-taste. Zubeachten ist dabei die Trägheit, d.h. das Raumschiff reagiert nicht sofort, sondern mit Verzögerung. Mit S kann der Countdown unterbrochen werden, mit der Spacetaste oder mit dem Feuerknopf wird er abgekürzt.

Die Schwierigkeitsgrade können in folgenden Zeilen verändert werden:

Zeile 2

T% = Treibstoffmenge

Zeile 5106

V = Fallgeschwindigkeit

Zeile 5107

RX = seitliche Abdrift



Wurm VC-64

Der Wurm soll Futter (Herzchen) fressen, muß aber das Gift (Kreuzchen) liegen lassen. Er darf nicht gegen die Wand laufen oder sich selbst anbeißen.

Passiert dies, skelettiert er. Die Todesursache wird angezeigt. Steuerung über Joystick (Control-port 2) oder über die Tastatur mit den Buchstaben I, J, K, M.

Der Wurm startet bei der ersten eingegebenen Bewegungsrichtung. Das Anzeigen der Todesursache u. s. w. kann abgekürzt werden durch Betätigen des Feuerknopfes oder einer beliebigen Taste während des Skelettierens. Bei der Punktberechnung wird die

Länge des zurückgelegten Weges negativ berücksichtigt. Jeder Spieler hat 3 Würmer, wenn alles Futter gefressen ist, wird im nächsten Level die Zahl des Giftes und die Geschwindigkeit erhöht.

Dennis Merbach

```

0 REM      ***** "WURM-3" *****
1  JO=56320:FB=12:FU=1:FW=7:FH=2:FK=0:Y=9:SI=54272 GOTO50000
2  TS=100:GS=30:FS=50:LW=5:A$="I":B$="M":C$="J":D$="K":DIMP(1000)
3  PRINTCHR$(0); "J":POKE53281,FU:POKE53280,FB:W=3:LX=1:L=LW:T=TS:G=0S:F=FS
5  GOTO20000
100 FORA=0T039:POKE1064+A,102:POKE55336+A,Y:POKE55256+A,Y:POKE1984+A,102:NEXT
110 FORA=0T09208TEP40:POKE1064+A,102:POKE1103+A,102:POKE55336+A,Y:POKE55375+A,Y
120 NEXT:TZ=T
200 FORA=0TOL-1:P(A)=1690+A:POKE55962+A,FW:POKEP(A),160:NEXT:POKEP(0),01
210 FORA=1TOF
215 PF=INT(960*RND(1))+40:IFPEEK(1024+PF)<>32THEN215
220 POKE1024+PF,83:POKE55296+PF,FH:NEXT:FORA=1TOG
230 PG=INT(RND(1)*960)+40:IFPEEK(PG+1024)<>32THEN230
240 POKE1024+PG,43:POKE55296+PG,FK:NEXT
298 GETR$:IFR$<>" "THEN298
299 PRINT"■TUNKTE";S:PRINT"§",, "LEVEL";LX,"WUERMER";W
300 IFJ$="J"THEN310
301 GETR$:IFR$=" "THEN300
302 GOTO1000
310 IF(PEEK(JO)AND15)=15THEN310
315 POKE649,0:R=0:GOTO1010
999 GETR$
1000 IFR$=A$THENR=-40:GOTO1004
1001 IFR$=B$THENR=40:GOTO1004
1002 IFR$=C$THENR=-1
1003 IFR$=D$THENR=1
1004 FORA=LTO1STEP-1:P(A)=P(A-1):NEXT:P(0)=P(0)+R:IFPEEK(P(0))<>32THEN2000
1005 POKESI+P(0):FW:POKEP(0),01:POKEP(1),160:POKEP(L),32
1006 FORA=1TOT:NEXT:N=N+1:IFJ$="K"THEN999
1010 J=PEEK(JO):IFABS(R)<>40AND(JAND1)=2THENR=-40:GOTO1004
1011 IFABS(R)<>40AND(JAND2)=0THENR=40:GOTO1004
1012 IFABS(R)<>1AND(JAND4)=0THENR=-1:GOTO1004
1013 IFABS(R)<>1AND(JAND8)=0THENR=1
1014 GOTO1004
2000 PX=PEEK(P(0)):IFPX<>83THEN2020
2001 SP=SP+1:L=L+1:P(L)=0:S=S+10:PRINT"§§TUNKTE";S:IFSP=FTHENSP=0:GOTO3000
2002 IFTX>1THENTX=TZ-4
2003 GOTO1005
2020 IFPX=102THENP$(1)="§SICH DEN KOPF EINGERANNT":P$(2)="DU KNERDER!":GOTO5000
2030 IFPX=43THENP$(1)="§HAT IFT GEFRESSEN":P$(2)="QUERG" GOTO5000
2040 P$(1)="§SICH SELBST ANGEKNABBERT":P$(2)="DU KNERDER!":GOTO5000
3000 POKEP(0),01:POKEP(1),160:POKE54272+P(0),FW
3001 FORA=1TO1000:NEXT
3005 POKE53281,13:PRINTCHR$(14);"WURM-3 WIE HABEN DEN WURM GEFUETTERT!"
3010 FORA=1TO(N/3):S=S-1:PRINT"WURM-3 TUNKTE";S;"|| "":NEXT:N=0
3015 FORA=1TO1000:NEXT
3020 LX=LX+1:SP=0:T=T/2:G=G+15:GOTO6010
4000 FORA=1TO2000:NEXT:POKE53281,0
4005 PRINTCHR$(14);"WURM-3 ALLE QUERMER SIND TOT!"
4010 FORA=1TOINT(N/3):S=S-1:PRINT"WURM-3 TUNKTE";S;"|| "":NEXT
4015 N=0:PRINT,"§HOECHSTPUNKTZAHL";HI
4020 POKE649,9:PRINT"WURM-3 NOCH EIN SPIEL (Y/?)"
4025 GETN$:IFN$<>" "THEN4035
4030 GETN$:IFPEEK(JO)<>127THENN$="J"
4031 IFN$=" "THEN4030
4033 IFN$<>"N"THENPRINTCHR$(142):GOTO4040
4035 PRINTCHR$(9);CHR$(142);"J":PRINT"VOR AUFNEHMEN RUNSTOP/RESTORE DRUECKEN":EN
    
```

```

J
4040 IFHI<STHENHI=S
4050 SP=0:S=0:GOTO3
5000 FORA=1TO800:NEXT:POKE51+P(A),0:POKEP(A),87
5010 FORA=1TOL:GETX$:IFS0=1THENNEXT:S0=0:GOTO6000
5015 IFPEEK(J0)<>127ORPEEK(203)<>64THEN80=1
5020 FORB=1TO100:NEXT:POKE51+P(A),B:POKEP(A),91:POKEP(A),87:NEXT
5030 FORA=1TO1000:NEXT:POKE53281,0:PRINTCHR$(14)"XXXXXXXXXXXXXXXX TER OCRM HAT ";
5040 PRINTP$(1);FORA=1TO1500:NEXT:PRINT"J":FORA=1TO200:PRINTP$(2);:FORB=1TO15
5050 NEXT:NEXT:FORA=1TO1000:NEXT
6000 W=W-1:IFW=0THEN4000
6010 POKE53281,FU:PRINTCHR$(142);"J":L=LW:SP=0:GOTO100
20000 POKE649,9:PRINTCHR$(14);"J"           LITTE WAELHEN           "
20010 PRINT,"XXXXXXXXX1  *PIELBEGINN"
20013 PRINT,"X X ":PRINT,"X2  OECHEL  JOYSTICK/KEYBOARD"
20015 PRINT,"X X ":PRINT,"X3  OERTE AENDERN"
20016 PRINT,"X X ":PRINT,"X4  TIDE"
20019 PRINT,"XXXXXITTE ZUMMER EINGEBEN"
20020 GETN$:IFPEEK(J0)<>127THEN20031
20021 IFN$=""THEN20020
20025 IFN$="1"ORN$="2"ORN$="3"ORN$="4"THENA=VAL(N$):GOTO20030
20026 GOTO20020
20030 ONAGOTO20031,20032,20040,4035
20031 PRINTCHR$(142);"J":L=LW:F=FS:G=GS:T=TS:GOTO100
20032 IFJ$="J"THENJ$="K":GOTO20000
20033 IFJ$="K"THENJ$="J":GOTO20000
20040 PRINT"J"           LITTE WAELHEN           "
20050 PRINT,"XXXXXXXXX1  ANZAHL DES LIFTS":PRINT,"X X ":PRINT,"X2  ANZAHL DES LUTTERS"
20060 PRINT,"X X ":PRINT,"X3  GESCHWINDIGKEIT":PRINT,"X X "
20070 PRINT,"X4  RICHTUNGSSTEUERUNG":PRINT,"X X ":PRINT,"X5  OURLAENGE"
20073 PRINT,"X X "
20075 PRINT,"X X ":PRINT,"X6  OERTE IN FRONUNG"
20080 GETN$:IFN$=""THEN20080
20090 IFN$="1"ORN$="2"ORN$="3"ORN$="4"ORN$="5"ORN$="6"THEN20100
20091 GOTO20080
20100 A=VAL(N$):ONAGOTO20120,20140,20160,20180,20200,20000
20120 PRINT"XXXXXXXXXANZAHL DES LIFTS:",A:ALTER OERT:;GS:PRINT,,"/EUER OERT";
20130 INPUTGS:GOTO20040
20140 PRINT"XXXXXXXXXANZAHL DES LUTTERS:",A:ALTER OERT:;FS:PRINT,,"/EUER OERT";
20150 INPUTFS:GOTO20040
20160 PRINT"XXXXXXXXXGESCHWINDIGKEIT:",A:ALTER OERT:;TS:PRINT,,"/EUER OERT: ";
20170 INPUTTS:GOTO20040
20180 PRINT"XXXXXXXXXRICHTUNGSSTEUERUNG: ALTE OERTE:":PRINT,"XXXXXUFWAERTS",A$
20181 PRINT,"XXXXXUWAERTS",B$:PRINT,"XLINKS",C$:PRINT,"XRECHTS",D$
20182 PRINT,"XNEUE OERTE:":PRINT,"XXXXXUFWAERTS",
20183 GETA$:IFA$=""THEN20183
20184 PRINTA$:PRINT,"XXXXXUWAERTS",
20185 GETB$:IFB$=""THEN20185
20186 PRINTB$:PRINT,"XLINKS",
20187 GETC$:IFC$=""THEN20187
20188 PRINTC$:PRINT,"XRECHTS",
20189 GETD$:IFD$=""THEN20189
20190 PRINTD$:FORA=1TO1000:NEXT:GOTO20040
20200 PRINT"XXXXXXXXXOURLAENGE:",A:ALTER OERT:;LW:PRINT,,"/EUER OERT (MAX10)";
20210 INPUTL:GOTO20040
50000 PRINT"XXXXXXXX", "  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z  "
50010 PRINT,"XXXXXXXXXIIII WURM  IIIII":PRINT,"          XXXXX"
50015 B=-2.7:FORA=14TO9STEP-1:B=B+2.79
50020 PRINTTAB(A)"EIN":FORC=1TOB:PRINT"  ";:NEXT:PRINT"SPIEL":
50030 FORC=1TO200:NEXT:IFAC14THEN50050
50040 PRINT,"X          VON":PRINT,"  DENNIS MERBACH,III":FORC=1TO2000:NEXT
50050 NEXT:FORA=1TO1000:NEXT:PRINTTAB(13)"GROSSARTIGES":FORA=1TO2000:NEXT
50060 PRINT"XXXXXXXXX          JOYSTICK ODER KEYBOARD (J/K)"
50070 GETJ$:IFPEEK(J0)<>127THENJ$="J":GOTO2
50075 IFJ$=""THEN50070
50080 IFJ$="J"ORJ$="K"THEN2
50090 GOTO50070
READY.

```



```

2502 IF<JAND16>=0ANDSW<=0ANDSY=221THENSW=1
2503 IFSW>0THENSY=SY-16:SW=SW-.06
2504 IFSW<=0THENSW=0:SY=221: SX=PJ
2506 GETA$: IFA$><"V" THEN2512
2510 GOTO2510
2512 IFPJ>2360RPJ<24THEN5000
2513 POKE53249,SY:POKE53278,0:POKE53248,SX:POKE53250,PJ:C=C+.015
2514 SC=PEEK(53278):IF<SCAND8>=8THENP3=40:SW=0
2515 P3=P3+1.6+C:POKE53255,P3:IFP3>=200THENA=3:GOTO6000
2516 IF<SCAND16>=16THENP4=40:SW=0
2517 P4=P4+1.3+C:POKE53257,P4:IFP4>=200THENA=4:GOTO6000
2518 IF<SCAND32>=32THENP5=40:SW=0
2519 P5=P5+2.0+C:POKE53259,P5:IFP5>=200THENA=5:GOTO6000
2520 IF<SCAND64>=64THENP6=40:SW=0
2521 P6=P6+1.6+C:POKE53261,P6:IFP6>=200THENA=6:GOTO6000
2522 IF<SCAND128>=128THENP7=40:SW=0
2523 P7=P7+1.2+C:POKE53263,P7:IFP7>=200THENA=7:GOTO6000
2524 PRINT"♣",MID$(TI$,4,1);": "RIGHT$(TI$,2):GOTO2500
5000 IFPJ<24THENPJ=24:GOTO2513
5010 T$=TI$:PRINT"♣♣♣♣ SIE HABEN SICH ERTRAENKT!"
5020 POKEVC+1,221:FORA=237TO255:POKEVC+2,A:POKEVC,A:NEXT
5030 FORA=221TO255:POKEVC+3,A:POKEVC+1,A:NEXT
5040 PU=VAL(MID$(T$,3,2))*60+VAL(RIGHT$(T$,2))
5050 PRINT,"♣♣♣♣SIE HABEN SICH":PRINT,PU"SEKUNDEN"
5055 PRINT"
GEHALTEN"
5060 PRINT"
ALTER REKORD:";HI;"SEKUNDEN":IFPU>HITHENHI=PU
5090 WAIT56321,16,127:FORA=3TO7:POKEVC+2*A,A*45-100:POKEVC+2*A+1,28:NEXT
5100 POKEVC,150:POKEVC+1,221:POKEVC+21,255:POKEVC+2,150:POKEVC+3,221:SY=221:SW=0
5110 FORA=1TO1500:NEXT:GOTO2000
6000 T$=TI$:P=PEEK(VC+A*2):POKEVC,PJ:POKEVC+1,221:FORB=202TO221:POKEVC+A*2+1,B
6010 NEXT:RG=-1:IFPJ>PTHENRG=1
6020 POKEVC+A*2,P
6030 IFP<>INT(PJ)THENP=P+RG:GOTO6020
6040 P=P+2:POKEVC+A*2,P:POKEVC+2,P:POKEVC,P
6050 IFP<254THEN6040
6060 PRINT"♣♣♣♣ DAS GESPENST HAT SIE ERTRAENKT!":GOTO5020
10000 POKEVC+21,15:POKEVC+23,15:POKEVC+29,15
10010 POKEVC,40:POKEVC+1,208:POKEVC+2,40:POKEVC+3,208:POKEVC+6,30:POKEVC+7,50
10020 POKEVC+16,4:POKEVC+4,20:POKEVC+5,50
10030 PRINTCHR$(14),"♣ LEISTERJAEGER"
10040 PRINT"♣♣♣♣ ♡IE IESPENSTER MOECHTEN ♡IE FANGEN!"
10050 PRINT"
XERSUCHEN ♡IE, SIE DARAN ZU HINDERN,"
10060 PRINT"
INDEM ♡IE SIE MIT DEM ♡PEER TREFFEN."
10070 PRINT"
♡IE KOENNEN SICH NACH LINKS UND NACH "
10090 PRINT"
RECHTS BEWEGEN, ENTWEDER MITTELS"
10100 PRINT"
'1' UND '2' ODER MITTELS ♡YSTICK,"
10110 PRINT"
UND SIE KOENNEN MIT DER ♡PACE'- "
10120 PRINT"
BZW MIT DER ♡EDERTASTE DEN ♡PEER":PRINT"
WERFEN."
10130 PRINT"♣
-ALLEN ♡IE NICHT IN DEN IRABEN!"
10140 PRINT,"
♣LERTIG?"
10150 GETA$: IFA$="" THEN10150
10160 PRINTCHR$(142);":":POKEVC+23,0:POKEVC+29,0:POKEVC+21,0:POKEVC,0:POKEVC+2,0
10170 GOTO113

```

READY.

64' VideoChips

VC-64-Sprites-Generator

Sprite-Grafik ist eine verhältnismäßig neue Form, um Zeichen und Figuren über den Bildschirm zu bewegen. Beim VC-64 lassen sich Sprites zusätzlich zum normalen Videobild auf dem Bildschirm darstellen.

Das Programm soll das Erstellen von Sprites erleichtern, indem es durch die Cursorsteuertasten und die Space-Taste in einer 24*21-Matrix die Kon-

struktion von Sprites erlaubt.

Nach Erstellung aller Sprites werden die entsprechenden Bytes in Datazeilen übernommen und das Hilfsprogramm gelöscht.

Erklärung des Programms:

- Zeile 20 Abfrage der Spriteanzahl (max. 4)
- Zeile 60 Abfrage der Speicherbereiche (11,13,14,15)
- Zeile 70 Abfrage der Zeilennummer ab der die Datazeilen gespeichert werden sollen (größer als 1000)
- Zeile 100 Ausgabe des Zeichenfeldes
Klammeraffe: Löschen des Spritepunktes
F1 übernimmt fertig gezeichnete Bytes in DATA-Zeilen
- Zeile 200 Sprites zeichnen und berechnen des Spritepunktes
- Zeile 400 Sprites übernehmen
- Zeile 650 Unterprogramm Punkte setzen/löschen
- Zeile 690 Löschmodul

Einschränkend sei vermerkt, daß maximal 4 verschiedene Sprites erstellt werden können, da mehr als 4 Sprites nicht problemlos abgespeichert werden können. Es dürfte aller-

dings kein Problem sein, das Programm entsprechend abzuändern. Außerdem können die Sprites nicht in Multicolor erstellt werden.

Rolf Werdehausen

```
10 PRINT"ERSTELLEN VON SPRITES!"
20 PRINT"WIEVIEL SPRITES";:INPUTSZ$
30 IF VAL(SZ$)<=0ORVAL(SZ$)>4THEN10
40 SZ=INT(VAL(SZ$)):PRINT"IN WELCHE SPEICHERPLAETZE SOLL GEPOKT WERDEN (<";
50 PRINT"FREI SIND: 11, 13,14 UND 15)?
60 PRINT:FORI=1TOSZ:PRINT"SPRITES NR" I " ";:INPUTSB:SB(I)=SB*64:NEXT
70 PRINT"DATA ZEILEN NR.: (>1000)";:INPUTDA$
80 IFVAL(DA$)<=1000ORVAL(DA$)>6500THENPRINT":":GOTO70
90 DA=INT(VAL(DA$))
100 DIMBS(21,3):I=0
110 POKE646,14:POKE198,0:I=I+1:IFI>SZTHEN690
120 PRINT"SPRITES NR: " I "
130 PRINT:FORZ=1TO21:PRINTTAB(11)ZTAB(15)"*****":NEXT
140 PRINT"ZEICHNEN
150 PRINT"CURSOR/SPACE
160 PRINT" LOESCHEN
170 PRINT"
180 PRINT" ENDE
190 PRINT"
200 SP=1:S=1:ZZ=1:BS=7
210 PO=1024+SP+14+(ZZ+2)*40:IFPEEK(PO)=160THENPE=1
220 ZE=170:FA=7:GOSUB650:GETT$
230 ZE=42:FA=14:GOSUB650:IFPE=1THENZE=160:FA=7:PE=0:GOSUB650
240 IFT$=""THEN210
250 REM SPRITES ZEICHNEN
270 IFT$="I"ORT$="" THEN340
280 IFT$="II"THEN360
290 IFT$="I"THEN430
300 IFT$="II"THEN450
310 IFT$="I"THEN470
320 IFASC(T$)=133THEN490
330 GOTO210
```

64' VideoChips

```
340 IFT$=" "THENIFPEEK(PO)<>160THENZE=160:FA=7:GOSUB650:BS(ZZ,S)=BS(ZZ,S)+2↑BS
350 IFSP=24ANDZZ<21THENZZ=ZZ+1:S=1:SP=1:BS=7:GOTO210
360 IFSP=24THEN210
370 SP=SP+1:BS=BS-1:GOTO400
380 IFSP=1THEN210
390 SP=SP-1:BS=BS+1
400 IFBS=-1THENBS=7:S=S+1:GOTO210
410 IFBS=8THENBS=0:S=S-1
420 GOTO210
430 IFZZ<>1THENZZ=ZZ-1
440 GOTO210
450 IFZZ<>21THENZZ=ZZ+1
460 GOTO210
470 IFPEEK(PO)<>42THENZE=42:FA=14:GOSUB650:BS(ZZ,S)=BS(ZZ,S)-2↑BS
480 GOTO210
490 REM SPRITES UEBERNEHMEN
500 FZ=DA:DA=FZ+INT((SZ+10)/10)*10
510 DA$=STR$(DA)+"DATA"
520 POKE646,6:PRINT"J"STR$(FZ)+"FORJ=0TO62:READQ:POKE"SB(I)+"J,Q:NEXT
530 D1=0:FORZZ=1TO21:FORS=1TO3
540 IFLEN(DA$)>70THENDA=DA+10:PRINTLEFT$(DA$,LEN(DA$)-1):DA$=STR$(DA+10)+"DATA"
550 DA$=DA$+RIGHT$(STR$(BS(ZZ,S)),LEN(STR$(BS(ZZ,S)))-1)+","
560 NEXT:NEXT
570 IFDA$=STR$(DA+10)+"DATA"THEN540
580 PRINTLEFT$(DA$,LEN(DA$)-1):DA=DA+10
590 PRINT"90DIMBS(21,3),SB("SZ"):I="I":SZ="SZ":DA="DA+10
600 PRINT"100":FORN=1TOSZ:PRINT"SB("N")="SB(N)">":NEXT:PRINT"II"
610 PRINT"500FZ="FZ+1
620 PRINT"GOTO90"
630 GOTO730
640 END
650 POKEPO,ZE:POKE54272+PO,FA:RETURN
690 ZA=10:ZE=120
700 POKE646,6:PRINT"J590ZA="ZE+10:ZE="ZE+110"
710 FORI=ZATOZESTEP10:PRINTI:NEXT:PRINT"GOTO690"
720 IFI>700THENPRINT"JPOKE646,14"
730 POKE631,19:FORI=1TO14:POKE631+I,13:NEXT:POKE198,15:END
```

IMTAL
COMPOSING
GMBH

Postfach 1207
3436 Hessisch Lichtenau
☎ Sa-Nr. 05602/4041
☒ 994026 vogt d

Fotosatz-Studio
Belichtungs-Service
Reproduktionen
Druckformherstellung
Plattenkopierstraße
Ganzseitenumbruch
über Bildschirm
Konturenprogramm
Daten-Fern-Übertragung
über Post-Modem
Weiterverarbeitung
Ihrer EDV-Daten zu Fotosatz

Disketten- schutz für Apple II (Teil 2)

Nachdem im letzten Heft ein Programm beschrieben wurde, mit dem Daten in einem nicht standardisierten Code auf Diskette aufgezeichnet wurden, wird hier nun ein Verfahren vorgestellt, das nur befugten Personen die Benutzung der Diskette erlaubt.

Das Wesentliche bei einem solchen Schutzsystem ist, daß der Benutzer keine Möglichkeit besitzen darf, in das Operationssystem (hier Applesoft-Basic mit DOS) zu kommen. Dies wird durch das Programm DOS PROTECTOR (Abb. 1) gewährleistet. Alle Abbruchbefehle wie RESET, CTRL-C, Erzwingen einer Fehlermeldung und auch IRQ und NMI werden von einem mit diesem Programm geänderten DOS mit einem „Aufhängen“ des Systems quittiert.

Um dieses Verhalten zu erreichen, werden im DOS auf Diskette einige Bytes geändert. So wird zum Beispiel der Programmteil „Fehlermeldung ausgeben“ so modifiziert, daß er die gleiche Aktion wie die RESET-Taste ausführt, nämlich den Sprung in eine Endlosschleife. Außerdem wird ein CTRL-C-Filter eingebaut. Wirkung: siehe oben.

Um nur befugtem Bedienungspersonal einen Zu-

griff auf Diskette zu gewährleisten, ist die Eingabe eines Codewortes sinnvoll. Ein konstantes Codewort allerdings schränkt dessen Funktion ein, da die Gefahr des „Abguckens“ doch relativ groß ist. Daher verlangt das Programm SPERRE, das in Abb. 2 zu sehen ist die Errechnung des einzugebenden Codezeichens. Eine 5 Zeichen lange Zeichenfolge, in der genau zwei Zahlen zwischen 0 und 4 vorkommen, wird angezeigt. Diese zwei Zahlen müssen addiert werden und das Ergebnis „geschiftet“ werden. Das heißt, die erhaltene Zahl muß zusammen mit der SHIFT-TASTE gedrückt werden. Eine Ausnahme bildet die Zahl 0: wenn sie als Summe erscheint, muß die Leertaste gedrückt werden. Als benutzerspezifischer Wert steht in Zeile 999 als DATA-Wert der Name der Diskette, der im oberen Bildschirmteil erscheint. Bei einer falschen Eingabe reagiert das Programm mit ... Aber lassen sie sich überraschen.

Das Programm SPERRE ruft seinerseits das Programm HELLO1 auf. Dieses Programm stellt ein einfaches Menüprogramm dar, das sich als Auswahl für bestimmte Programme eignet. Auch hier erscheinen die benutzerabhängigen Werte in DATA-Zeilen. In Zeile 999 wird der Name der Diskette abgelegt. Ab Zeile 1000 sind dann die einzelnen Programmnamen gespeichert. Als Zusatz zum Programmnamen wird noch eine Ziffer abgerufen, die die Art des Programmaufrufs spezifiziert. 1 steht dabei für BRUN, 2 für RUN und 3 für EXEC.

Die Programme sollen nun etwas genauer untersucht werden.

Zuerst DOS PROTECTOR: Diese Routine liest bestimmte Sektoren der Diskette ein, ändert einige Bytes und schreibt dann die geänderten Sektoren wieder auf die Diskette zurück. In den Zeilen 70, 110, 140, 170, 200 und 220 wird der einzulesende Sektor definiert und mit einem GOSUB 1040 gelesen.

Was bewirken jetzt aber die einzelnen POKES? In den Zeilen 70-100 wird der Teil des DOS geändert, der spezifiziert, welche Werte in den Speicherstellen 3D0-3FF stehen. In Zeile 90 werden diese Werte so umgestellt, daß die Speicherstellen von 3EF bis 3F1 und von 3F5 bis 3FF alle den gleichen Wert enthalten, nämlich ein JMP (\$03F2). Dieser Befehl bewirkt einen Sprung zu der Adresse, die in \$3F2 abgelegt ist. Genau das passiert aber auch bei einem RESET. Die Zeile 80 ändert den Teil des DOS um, der die RESET-Adresse er-

rechnet. Normalerweise wird nämlich in \$3F2 und \$3F3 die gleiche Adresse wie in \$3D1 und \$3D2 gespeichert. Anstelle dessen steht dort nun die absolute Adresse \$3F8. Da dort ein JMP (\$03F2) steht, führt ein Reset zur Endlosschleife.

Zeile 120 bewirkt, daß, wenn nach einem LOAD-Befehl, aber vor Rücksprung in das Basic CTRL-C gedrückt wird, dieser Tastendruck ignoriert wird. Er würde nämlich vom DOS nicht verarbeitet werden und somit auch nicht zur Endlosschleife führen. Zeile 180 ändert das Fehlermeldungsmodul, so daß ein Sprung in die Endlosschleife erfolgt.

Zeile 205 schließlich stellt den CTRL-C-Filter und den Unterprogrammteil, der CTRL-C nach einem LOAD-Befehl unterdrückt. Die Zeilen 150 und 230 sorgen dafür, daß dieser CTRL-C-Filter auch aktiviert wird.

Die Zeilen 1000-1035 stellen ein Unterprogramm dar, das den IOB (I/O-Control-Block) lokalisiert. Dieser IOB wird in den Unterprogrammen in den Zeilen 1040-1060 und 1070-1090 benutzt, um die RWTS (Read or Write a Track and Sector) zu steuern.

Nun zur Analyse des Programms SPERRE. In Zeile 1 werden zunächst zwei Zufallsfunktionen definiert. Die Funktion R1 liefert den ASCII-Wert eines zufälligen Zeichens zwischen 0 und 4, generiert also eine Zufallszahl. Die Funktion R2 hingegen generiert ein zufälliges Zeichen zwischen § und /. Zeile 4 ist sehr interessant. Hier wird nämlich in RL und RH der momentane RESET-Vector gespei-

chert. Anstatt dessen wird dort der Wert \$E199 gespeichert. An dieser Stelle steht die Applesoft-Fehlermeldung „Illegal Quantity Error“. Wichtig ist an dieser Stelle der Befehl CALL -1169. Warum, wird an späterer Stelle erläutert. In Zeile 5 steht nun der Befehl ONERR GOTO 210.

Durch die Änderung des RESET-Vektors wird also bei RESET die oben genannte Fehlermeldung ausgegeben. Diese wird aber vom ONERR GOTO-Befehl abgefangen und führt zu einem Sprung nach Zeile 210. Damit wurde also der Befehl ON RESET GOTO 210 simuliert. Die Zeilen 10-40 führen zur Ausgabe des oberen Bildschirmteils. Wichtig sind wieder die Zeilen 50-70. Diese Zeilen entstanden aus der Notwendigkeit, die Zufallsfunktion RND(1) wirklich zufällig zu machen. Denn das ist sie keinesfalls. Dieses kann mit Hilfe eines kleinen Programms leicht bewiesen werden, das als HELLO-Programm auf Diskette abgelegt wird, und Zufallszahlen erzeugt. Nach dem Kaltstart des Systems werden dort immer die gleichen Zahlen erscheinen.

Daher warten die Zeilen 50-70 des Programms SPERRE auf einen Tastendruck. So lange dieser nicht erscheint werden „Dummy-“ Zufallszahlen erzeugt.

Die Zeilen 95-180, die durch die FOR-NEXT Schleife in Zeile 90 insgesamt 5 mal durchlaufen werden, fordern den Benutzer zur Eingabe eines Codezeichens auf. Da diese Zeilen relativ komplex aufgebaut sind, würde eine Analyse hier zu weit führen und bleibt dem Leser selbst überlassen.

Die Zeilen 210-310 werden bei einer Fehleingabe durchlaufen. Das Ergebnis soll hier nicht verraten werden, sondern selbst ausprobiert werden.

In Zeile 500 wird wieder der alte RESET-Vektor zurückgeschrieben. Der CALL 1002 ist zur Reaktivierung des DOS unbedingt notwendig.

Der Befehl CALL -1169, der nach jeder Modifikation des RESET-Vektors auftaucht hat folgenden Sinn: Um unterscheiden zu können, ob der Rechner gerade einen Einschalt-Reset ausführt oder einen Lauf-Reset, wird in der Speicherstelle \$3F4 ein bestimmter Wert abgelegt, der vom Autostart-Monitor-ROM abgefragt wird. Sollte dieser Wert vorhanden sein, so wird ein JMP (\$03F2) ausgeführt, andernfalls ein Cold-Boot. Um diesen Cold-Boot zu verhindern wird der Befehl CALL -1169 verwendet, der diesen bestimmten Wert in \$3F4 schreibt. Da dieser Wert nicht konstant ist, muß er nach jeder Modifikation des RESET-Vektors neu errechnet werden. Näheres dazu findet sich in (1)-(6).

Das Programm HELLO1 ist sehr einfach aufgebaut. Auch hier findet der ONERR GOTO-Befehl Verwendung. Er wird benutzt, um das Ende der Data-Zeilen zu erkennen. In Zeile 10-55 wird der Bildschirm erstellt. Die Zeilen 60-80 geben die in den DATA-Zeilen gespeicherten Programmnamen mit einem vorausgestellten Kennbuchstaben aus. Dabei ist anzumerken, daß eine Leere DATA-Zeile eine Leerzeile auf dem Bildschirm bewirkt (siehe Zeile 70). In Zeile 99 wird der ONERR GOTO-Befehl

wieder gelöscht, und es wird die Eingabe des Kennbuchstabens vom aufzurufenden Programm verlangt. Nach dessen Überprüfung (Zeile 120) wird der entsprechende Programmname auf dem Bildschirm mit blinkender Schrift gekennzeichnet (Zeilen 140-185) und das Programm aufgerufen (Zeilen 190-210).

Eine optimal geschützte Diskette würde also das Bootprogramm das Programm HELLO von der SYSTEM MASTER-Diskette enthalten. Dieses Programm würde dann das Programm SPERRE aufrufen. Von dort aus wird dann das Menüprogramm HELLO1 aktiviert.

Wichtig: Aus den im ersten Absatz geschilderten Bedingungen darf natürlich kein Programm mit dem END-Befehl aufhören. Dieser Befehl bewirkt nämlich genau das, was mit dem Schutzsystem verhindert werden soll: den Ausstieg in die Kommandoebene des BASIC. Der Befehl END sollte ersetzt werden durch CALL 1016. Damit wird ein Sprung in die Endlosschleife erreicht.

Der Autor möchte noch erwähnen, daß dieses System keineswegs absolut knacksicher ist. Es ist aber eine einfache Möglichkeit, nur befugten Personal Zugriff zu bestimmten Disketten zu gewähren. Das typische Vorgehen beim Erstellen einer geschützten Diskette ist wie folgt:

Zuerst, wie im DOS-Manual beschrieben, das Standard-Hello-Programm von der System-Master-Diskette so modifizieren, daß das Programm SPERRE aufgerufen wird, nachdem die Language-Card

geladen wurde. Der verbleibende END-Befehl ist zu ersetzen durch CALL 1016. Mit diesem Programm als Boot-Programm die Disketten nun initialisieren. Jetzt die Programme SPERRE und HELLO1 auf die neu initialisierte Diskette kopieren. Dabei jeweils den in Zeile 999 abgelegten Diskettennamen ändern. Im Programm HELLO1 sind daraufhin in die DATA-Zeilen ab 1000 die wählbaren Programme einzutragen (Bitte nicht die Codeziffer für den Aufruftyp vergessen). Eine leere DATA-Zeile bewirkt dabei eine Leerzeile auf dem Bildschirm. Als letztes werden die Programme auf die Diskette kopiert.

Bei Basic-Programmen ist dabei jeder END-Befehl durch CALL 1016 zu ersetzen. Bei Maschinenprogrammen ist das Vorgehen wesentlich komplizierter. Diese Beschreibung würde den Rahmen dieses Artikels sprengen.

Damit ist die Diskette bootfertig. Um sie nun zu schützen, muß zuerst das Programm DOS PROTECTOR aufgerufen werden, mit dem das DOS modifiziert wird. Danach kann das Programm SPECIAL DOS aktiviert werden, mit dem die Daten im Spezialformat aufgezeichnet werden. Damit ist die Diskette geschützt und fertig für die Verwendung.

Vorschläge und Anregungen zu diesen Programmen richten Sie bitte an die Redaktion.

Jörg Bliesener


```

10 TEXT : HOME : A$ = "
    S P R O T E C T O R
    " : A$A$ : N O R M A L
20 VTAB 12: HTAB 1: PRINT "DRIVE ? (1/2) ->";
30 GET A$: IF A$ < > "1" AND A$ < > "2" THEN 30
40 DR = VAL (A$)
50 GOSUB 1000
60 VTAB 12: HTAB 1: PRINT "DOS PROTECTION RUNNING..."
70 T = 0: S = 13: GOSUB 1040
80 POKE 560,169: POKE 561,3: POKE 562,234: POKE 571,169: POKE 572,248: POKE 573,234
90 POKE 625,248: POKE 626,3: POKE 630,108: POKE 631,242: POKE 632,35: POKE 633,108: POKE 634,242
: POKE 635,3: POKE 636,108: POKE 637,242: POKE 638,3: POKE 639,242: POKE 640,3
100 GOSUB 1070
110 T = 1: S = 3: GOSUB 1040
120 POKE 569,76: POKE 590,105: POKE 591,186
130 GOSUB 1070
140 T = 0: S = 0: GOSUB 1040
150 POKE 575,32: POKE 576,208: POKE 577,8: POKE 720,169: POKE 721,111: POKE 722,133: POKE 723,5
6: POKE 724,169: POKE 725,186: POKE 726,133: POKE 727,57: POKE 728,96
160 GOSUB 1070
170 T = 1: S = 5: GOSUB 1040
180 POKE 725,108: POKE 726,242: POKE 727,3
190 GOSUB 1070
200 T = 0: S = 4: GOSUB 1040
205 FOR I = 617 TO 633: READ A: POKE I,A: NEXT
210 GOSUB 1070
220 T = 0: S = 1: GOSUB 1040
230 POKE 580,32: POKE 581,208: POKE 582,8: GOSUB 1070
240 VTAB 12: HTAB 1: PRINT "DOS PROTECTION COMPLETED."
250 END
10000 DATA 32,227,3,133,7,133,6,96,165,7,164,6,32,217,3,96
1010 FOR I = 768 TO 783: READ J: POKE I,J: NEXT
1020 CALL 768
1030 I08 = PEEK (6) + 256 * PEEK (7)
1040 POKE I08 + 2,DR: POKE I08 + 3,0: POKE I08 + 4,T: POKE I08 + 5,S: POKE I08 + 8,0: POKE I08
+ 9,2: POKE I08 + 12,1: POKE I08 + 13,0
1050 CALL 776
1060 RETURN
1070 POKE I08 + 2,DR: POKE I08 + 3,0: POKE I08 + 4,T: POKE I08 + 5,S: POKE I08 + 8,0: POKE I08
+ 9,2: POKE I08 + 12,2: POKE I08 + 13,0
1080 CALL 776
1090 RETURN
10900 DATA 173,16,192,108,96,157,32,27,253,201,131,208,3,76,16,164,96
    
```

Abb. 1: DOS PROTECTOR

```

1 DEF FN R1(X) = INT ( RND ( 1 ) * 5 ) + 48: DEF FN R2(X) = INT ( RND ( 1 ) * 48 ) + 64
4 RL = PEEK ( 1010 ) : RH = PEEK ( 1011 ) : POKE 1010,153: POKE 1011,225: CALL - 1169
5 ON LR GOTO 210
10 HOME : FOR I = 1 TO 40: A$ = A$ + " " : NEXT
30 READ N$
40 INVERSE : PRINT A$: SPC ( 20 - LEN ( N$ ) / 2 ) : SPC ( 40 - POS ( 0 ) ) : A$ : A$: N O R M A L
50 PRINT : PRINT : VTAB 14: HTAB 15: PRINT "PRESS ANY KEY"
60 POKE - 16368,0
70 IF PEEK ( - 16364 ) < 128 THEN X = RND ( 1 ) : GOTO 70
80 POKE - 16368,0: VTAB 6: CALL - 958
90 FOR I = 1 TO 5: VTAB 12: HTAB 18
95 X(1) = FN R1(3) : X(2) = FN R1(4) : C$(1) = CHR$ ( FN R2(3) ) : C$(2) = CHR$ ( FN R2(12) ) : C$(3)
= CHR$ ( FN R2(2) )
100 P(1) = INT ( RND ( 1 ) * 4 ) + 1
110 P(2) = FN R1(34) - 47: IF P(2) < = P(1) THEN 110
120 CX = 1: CC = 1
130 FOR J = 1 TO 5
140 IF J = P(CX) THEN PRINT CHR$ (X(CX)) : CX = CX + 1
    
```

Abb. 3: HELLO!

```

150 IF J < > P(CX) THEN PRINT C$(CX) : CC = CC + 1
160 NEXT
170 X$ = CHR$ ( X( 1 ) + X( 2 ) - 64 )
180 VTAB 15: HTAB 3: PRINT "CODE CHARACTER ->"; POKE - 16368,0: WAIT - 16384,128: GET C$
190 IF C$ = X$ THEN NEXT
200 IF I = 6 THEN 500
210 RESTORE : READ N$: FOR J = 778 TO 798: READ I: POKE J,I: NEXT
230 FOR J = 1 TO 4: READ I: POKE 777,I: CALL 778: NEXT
270 HOME : FLASH : PRINT A$: SPC ( 17 ) : "*****"; SPC ( 35 ) : "*****"; SPC ( 35 ) : "*****"; SPC ( 25 ) : "
*****"; SPC ( 15 ) : "*****"; SPC ( 15 ) : "*****"; SPC ( 15 ) : "*****"; SPC ( 15 ) : "*****"
280 S$ = "
        " : FOR I = 1 TO 15: PRINT S$: NEXT
290 PRINT A$: FOR W = 1 TO 5000: NEXT
300 HOME : N O R M A L : VTAB 12: PRINT " E V E R Y T H I N G I S O V E R
310 POKE 1010,0: POKE 1011,3: CALL - 1169: POKE 768,76: POKE 769,0: POKE 770,3: CALL 768
500 POKE 1010,RL: POKE 1011,RH: CALL - 1169: CALL 1002: PRINT : PRINT CHR$ ( 4 ) "RUN HELLO!"
999 DATA "GAMES III"
1000 DATA 173,48,192,136,208,5,206,9,3,240,9,202,208,245,174,8,3,76,10
1005 DATA 3,96
1010 DATA 200,150,200,150,200,150,240,255
    
```

Abb. 2: SPERRE

```

10 HOME
20 FOR I = 1 TO 40: A$ = A$ + " " : NEXT
30 READ N$: INVERSE : PRINT A$ : " DOS 3,3" : SPC ( 29 - LEN ( N$ ) ) : N$ : " : A$:
40 FOR I = 1 TO 20: PRINT A$: NEXT
50 VTAB 4: PRINT "WELCHES PROGRAMM ?"
55 PRINT : PRINT
60 I = 1: ON ERR GOTO 99
70 READ A$: IF A$ = " " THEN PRINT : GOTO 70
7: READ C
75 PRINT CHR$ ( I + 64 ) : " : A$
80 I = I + 1: GOTO 70
99 POKE 216,0
100 VTAB 22: PRINT "KENNBUCHSTABE EINGEBEN.." : SPC ( 25)
105 I = I - 1
110 GET A$
120 IF A$ < "A" OR A$ > CHR$ ( I + 64 ) THEN 110
130 HTAB 25: PRINT A$
135 B$ = A$
140 RESTORE : READ A$
150 FOR I = 1 TO ASC ( B$ ) - 64
160 READ A$
165 IF A$ = " " THEN E = E + 1: GOTO 160
167 READ C
170 NEXT
180 VTAB 5 + I + E: FLASH : HTAB 5: PRINT A$
185 N O R M A L
190 IF C = 1 THEN PRINT CHR$ ( 4 ) "BRUN " : A$
200 IF C = 2 THEN PRINT CHR$ ( 4 ) "RUN " : A$
210 IF C = 3 THEN PRINT CHR$ ( 4 ) "EXEC " : A$
999 DATA GAMES 111
1000 DATA GAME NO. 1,2
1020 DATA GAME NO. 2,2
1030 DATA SUPER GAME NO. 1,1
1040 DATA EXEC FILE NO. 1,3
    
```

Trollhöhlen

Obwohl für den Apple geschrieben, paßt das Programm in jeden Rechner. Die Befehle HOME und

VTAB bedeuten: 'Lösche Bildschirm' und 'Vertikal-Tabulator'.

```

50 HOME : VTAB 5
52 PRINT "HOEHLE NR. 1": PRINT
55 PRINT "TROLL: 'DU DARFST DIR SOVIELE KUPFER-          MUENZEN NEHMEN WIE DU WILLST.
    DAZU SCHENKE ICH DIR SCHLUESSEL          NR 1.'"
60 PRINT : INPUT "TROLL: 'ALSO WIEVIELE MUENZEN WILLST DU          NEHMEN?' ";A$
65 KM = VAL (A$):BM = 0:S = 1
100 HOME : VTAB 5
102 PRINT "HOEHLE NR. 2": PRINT
110 PRINT "TROLL: 'DU BEKOMMST SOVIELE KUPFERMUEN-          ZEN WIE DU BRONZEMUENZEN HAST
    UND UMGEKEHRT.'"
120 KM = KM + BM:BM = KM
130 GOSUB 5000
200 HOME : VTAB 5
202 PRINT "HOEHLE NR. 3": PRINT
210 PRINT "TROLL: 'WENN DU WENIGER ALS 3 MUENZEN          HAST, DANN SCHMEISSE ICH DICH IN
    EINE BODENLOSE GRUBE'"
220 IF BM + KM < 3 THEN 300
230 GOSUB 5000
290 GOTO 400
300 GOSUB 5000
310 HOME
320 END
400 HOME : VTAB 5
402 PRINT "HOEHLE NR. 4": PRINT
410 PRINT "TROLL: 'ICH NEHME MIR ZWEI BRONZEMUEN-          ZEN VON DIR!'"
420 BM = BM - 2: IF BM < 0 THEN BM = 0
430 GOSUB 5000
500 HOME : VTAB 5
502 PRINT "HOEHLE NR. 5": PRINT
510 PRINT "TROLL: 'ICH JAGE DICH ZURUECK, WENN DU          MEHR ALS EINE BRONZEMUENZE BEI
    DIR HAST.'"
520 GOSUB 5000
530 IF BM > 1 THEN 400
600 HOME : VTAB 5
602 PRINT "HOEHLE NR. 6": PRINT
610 PRINT "TROLL: 'WENN DU KEINE BRONZEMUENZEN HAST          DANN MUSST DU DIE TUER ZU DER
    DEIN SCHLUESSEL PASST DEFFNEN!'"
620 GOSUB 5000
630 IF BM > < 0 THEN 700
640 IF S < 5 THEN 800
650 HOME : VTAB 12
652 PRINT "HOEHLE NR. 7": PRINT
660 PRINT "ZAUBERER: 'WENN DU MEHR ALS ZWEI KUPFER-          MUENZEN HAST VERWANDELE ICH
    DICH IN EINEN WURM!'"
670 GOSUB 5000
680 HOME : VTAB 12: IF KM > 2 THEN PRINT "DU BIST EIN WURM.": END
690 PRINT "DU HAST ES GESCHAFFT...": END
700 HOME : VTAB 5
702 PRINT "HOEHLE NR. 9": PRINT
710 PRINT "TROLL: 'ICH TAUSCHE DEINEN SCHLUESSEL          GEGEN DEN NAECHST HOEHEREN.
    WENN DU MIR SCHLUESSEL NR.5          GIBST, DANN FRESSE ICH DICH.'"
720 IF S < 5 THEN S = S + 1: GOSUB 5000: GOTO 900
730 GOSUB 5000
740 PRINT "TROLL: 'ICH HABE DICH VIERMAL GEWARNT!'"
750 END
800 HOME : VTAB 5
802 PRINT "HOEHLE NR. 8": PRINT
810 PRINT "DU BIST IN EINE HOEHLE VON MENSCHENFRES-          SENDEN BANDWUERMERN GERATEN UND WIRST GERA

```


Apple-Kiste

```

DE VERSPEIST.": END
900 HOME : VTAB 5
902 PRINT "HOEHLE NR. 10": PRINT
910 PRINT "TROLL: 'ICH TAUSCHE DEINE BRONZEMUENZEN          GEGEN KUPFERMUENZEN.'"
920 KM = KM + BM:BM = 0
930 GOSUB 5000
1000 HOME : VTAB 5
1002 PRINT "HOEHLE NR. 11": PRINT
1010 PRINT "TROLL: 'ICH GEBE DIR EINE BRONZEMUENZE.'"
1020 BM = BM + 1
1030 GOSUB 5000
1100 HOME : VTAB 5
1102 PRINT "HOEHLE NR. 12": PRINT
1110 PRINT "TROLL: 'ICH NEHME MIR ZWEI KUPFERMUEN-          ZEN.'"
1120 KM = KM - 2: IF KM < 0 THEN KM = 0
1130 GOSUB 5000
1200 HOME : VTAB 5
1202 PRINT "HOEHLE NR. 13": PRINT
1210 PRINT "TROLL: 'DU DARFST NACH OBEN, WENN DU          KEINE KUPFERMUENZEN MEHR HAST.
ANDERNFALLS GEHT ES DURCH DEN          ZWEITEN AUSGANG ZURUECK!'"
1230 GOSUB 5000
1240 IF KM = 0 THEN 100
1250 GOTO 1000
4999 END
5000 PRINT : PRINT "KUPFERMUENZEN: ";KM
5010 PRINT "BRONZEMUENZEN: ";BM
5020 PRINT "SCHLUESSEL NR.: ";S
5030 PRINT : PRINT "DRUECKE EINE TASTE. ";: GET A$: RETURN
    
```



EXPLORING SPECTRUM BASIC £4.95 An ideal complement to the Sinclair manual. Over 50 games, application & utility programs plus explanations of programming techniques. Mike Lord. 192 pages.

THE EXPLORERS GUIDE TO THE ZX81 £4.95 The book for the ZX81 enthusiast, now in edition 2 with 152 pages of games, engineering & utility programs plus much useful information on machine language and hardware. Mike Lord.

WAKE UP YOUR ATOM £4.95 20 great programs to make the most of your Atom; including the colour board (old or new type) if fitted plus copious programming tips. Brian Lloyd.

THE ATOM MAGIC BOOK £5.50 A wealth of games and other programs plus much useful software and hardware information. Mike Lord.

Also available

MASTERING MACHINE CODE ON YOUR ZX81 Tom Baker 180 pages £7.50

GETTING ACQUAINTED WITH YOUR ZX81 Tim Hartnell £4.95

GETTING ACQUAINTED WITH YOUR ACORN ATOM Tim Hartnell & Trevor Sharples £7.95

PRACTICAL PROGRAMS FOR THE BBC COMPUTER & ACORN ATOM D. Johnson Davies £5.95

WHAT CAN I DO WITH 1K? Roger Valentine £4.95

ATOM RAM BOARDS SAE for details

ATOM ROM BOARDS £35.00 inclusive. Software switch between 3 utility ROMs and 4K RAM (fitted) to load your own utilities from tape or disc.

All prices include UK P&P and VAT where applicable. Overseas customers add £1.50 per item for surface air.

TIMEDATA Ltd. Dept HC 1
16 Hemmells, Laindon, Basildon, Essex Tel: (0268) 418121



TIMEDATA

Computerhirn = Mastermind

Für den CBM stellt Homecomputer heute 2 Mastermind-Versionen vor. In der einen erhält der Spieler die Option, selbst den versteckten Code zu finden, oder dem Computer die Aufgabe zu stellen! In der anderen spielen zwei Spieler gleichzeitig gegeneinander und jeder kann das Ergebnis des Gegners mit auswerten, um die Lösung zu finden.

Mastermind 1

Nachdem man das Mastermindprogramm eingetastet hat, darf man nicht gleich mit RUN starten! Zuerst wird RUN 2000 ein-

gegeben. Dabei wird erst einmal überprüft, ob man die Zeilen 0 bis 5 exakt eingegeben hat. Das ist nötig, da dort das Maschinenpro-

gramm hingePOKEd wird. Wenn alles in Ordnung ist, dann kann man das Programm starten.

Wenn der Computer die Zahl raten soll, dann wendet er folgende Strategie an:

Zuerst nimmt er Zufallszahlen als Versuch, bis er alle Ziffern einmal probiert hat; höchstens jedoch drei mal. Daß er alle in der Lösung vorkommenden Ziffern einmal probiert hat, merkt er daran, daß die Summe von allen Schwarzen und Weißen gleich Drei ist. Dabei kommt keine Ziffer doppelt vor, weil Ziffernkombinationen mit Ziffern, die schon einmal vorkamen, am Anfang nicht so informativ sind. Für den Menschen ist es vielleicht einfacher, wenn er auch schon am Anfang Ziffern wiederholt verwendet, weil er dann nicht so viel denken muß. Den Computer stört das nicht.

Falls unter den ersten drei Versuchen keine Ziffer doppelt vorkommt, weiß der Computer nach drei Versuchen, wenn man ihm noch nicht drei Schwarze

und Weiße gegeben hat, daß die noch nicht probierte Ziffer in der Lösung vorkommt.

Dann wird erst das Maschinenprogramm benutzt. Und zwar wendet es folgende Strategie an: Es zählt von 000 bis 999. Bei jeder Ziffernkombination geht es alle Versuche durch und berechnet bei jedem Versuch, wieviel Schwarze und Weiße es hätte erhalten müssen, wenn die angenommene Kombination die Lösung wäre.

Wenn es bei allen bisherigen Versuchen genauso viele Schwarze und Weiße berechnet, wie es auch bekommen hat, so verwendet es die laufende Ziffernkombination als nächsten Versuch, d.h., daß es als jeweils nächsten Versuch die kleinste Zahl nimmt, bei der die Annahme, daß es sich um die Lösung handelt, zu keinem Widerspruch zu den bisherigen Versuchen führt.

Die Versuche werden nicht in dimensionierten Variablen, sondern in den zweiten Kassettenbuffer abgelegt.

```

0 REM COPYRIGHT BY CORNELIUS GEPPERT
1 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
2 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
3 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
4 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
5 REMCOMPUTER
70 GOTO300
75 R%=10*RAND(1):IFZ(R%)THEN75
77 Z(R%)=1:RETURN
80 POKE285,N:SYS1068
100 A%=PEEK(269):B%=PEEK(268):C%=PEEK(267)
120 IFPEEK(271)=0THENPRINT:PRINTTAB(7)"ERRAPE HUMANUM EST":GOTO1250
240 N=N+5
250 IFETHEN1140
260 GOTO430
280 END
300 PRINT"☐ ☐ ☐ HOMECOMPUTER - M A S T E R - M I N D "
305 PRINT"☐☐☐ WOLLEN SIE DIE ZAHL RATEN? ";
310 GETB$:IFB$=""THEN310
312 GOSUB1400 A$=B$
320 PRINT:PRINT:IFA$="N"THENPRINT" SOLL DER COMPUTER GEGEN":PRINT:PRINT" SICH
SELBER SPIELEN? ";
330 IFA$="N"THENWAIT158,1:GETB$:GOSUB1400
338 N=0:POKE280,1
340 FORN=267TO270:POKEM,N:NEXT
342 FORN=1TO1E3:NEXT:PRINT"☐
345 IFA$="J"THENE=0:GOTO1020
347 IFB$="J"THENE=1:GOTO1020

```



```

360 GOSUB75:AX=FX:GOSUB75:BX=RX:GOSUB75:CX=RX
425 IFE=1THEN1140
427 IFN=0THENPRINT"NOHVERSUCH!"
430 PRINT:PRINTAXBMCX
440 PRINT:INPUT" WIEVIELE SCHWARZE UND WEISSE ";S,W:SU=SU+S+W
445 IFS+W>3THENPRINT"MAN MUESSTE MASTER-MIND SPIELEN KOENNEN!":GOTO440
447 POKE896+N,AX:POKE897+N,BX:POKE898+N,CX:POKE899+N,S:POKE900+N,W
450 GOSUB1300
480 IFSUC3ANDNC10THENN=N+5:GOTO360
490 IFS<3THEN00
500 PRINT:PRINT"ICH HABE NUR"N/5+1"VERSUCH"LEFT$( "E",1ANDN>0)" GEBRAUCHT":GOTO12
50
1020 A=INT(10*RND(1)):B=INT(10*RND(1)):C=INT(10*RND(1))
1030 IFE=1THEN360
1040 INPUT" WIE LAUTET DER ERSTE VERSUCH "V:GOTO1110
1050 GOSUB1300
1070 IFE=0THENN=N+5
1075 IFSUC3ANDNC10ANDETHENN=N+5:GOTO360
1080 IFS<3ANDETHEN00
1085 IFETHEN500
1086 IFS<3THEN1090
1087 PRINT:PRINT"SIE HABEN NUR"N/5+1"VERSUCH"LEFT$( "E",1ANDN>0)" GEBRAUCHT":GOTO
1250
1090 INPUT" WIE LAUTET DER NAECHSTE VERSUCH"V
1110 AX=V/100+1E-4
1120 BX=10*(V/100-INT(V/100))+1E-3)
1130 CX=10*(V/10-INT(V/10))+1E-3)
1140 S1=1:S2=S1:S3=S1:S4=S1:S5=S1:S6=S1:S=0:W=5
1150 IFA=AXTHENS=S+1:S1=0:S2=S1
1160 IFB%=BTHENS=S+1:S3=0:S4=S3
1170 IFC%=CTHENS=S+1:S5=0:S6=S5
1180 IFA%=BANDS2ANDS3THENW=W+1:S2=0:S3=S2
1190 IFA%=CANDS2ANDS5THENW=W+1:S5=0
1200 IFA=B%ANDS1ANDS4THENW=W+1:S1=0:S4=S1
1210 IFB%=CANDS4ANDS5THENW=W+1
1220 IFA=C%ANDS1ANDS6THENW=W+1:S6=0
1230 IFC%=BANDS3ANDS6THENW=W+1
1235 POKE896+N,AX:POKE897+N,BX:POKE898+N,CX:POKE899+N,S:POKE900+N,W:SU=SU+S+W
1240 GOTO1050
1250 PRINT"NOCHMAL SPIELEN (LASSEN) ?
1260 WAIT150,1:GETB$:IFB$="J"THENRUN
1270 PRINT"SCHADE !"
1280 PRINT:PRINT" AUF WIEDERSEHEN"
1290 END
1300 IFEANDN>0THENFORN1=N1+5TON1STEP5:GOTO1330
1305 INPUT:PRINTTAB(22);"ANZAHL DER
1310 PRINT" VERSUCH"SPC(9)"SCHWARZEN UND WEISSEN
1320 FORN1=896TON+N1STEP5
1330 PRINTPEEK(N1)PEEK(1+N1)PEEK(2+N1),,PEEK(3+N1),PEEK(4+N1):NEXT:N1=N1-5
1340 RETURN
1400 PRINTLEFT$( "JA",2ANDB$="J")LEFT$( "NEIN",4ANDB$="N"):RETURN
2000 FORI=0TO5:READA:IFPEEK(A)ORPEEK(A+4)THENPRINT"DIE ZEILE"ISTIMMT NICHT.":EN
D
2010 NEXT
5000 DATA1061,1132,1203,1274,1345,1359
5010 FORI=1062TO1359:READA:POKEI,A:NEXT
5020 PRINT"JETZT KOENNEN DIE ZEILEN AB 2000
5030 PRINT"GELOESCHT WERDEN.
6000 DATA129,4,1,0,143,34,141,15,1,174,41,4,169,1,160,6,153,15,1,136,208,250,140
6010 DATA22,1,140,23,1,189,128,3,77,13,1,208,9,238,22,1,140,16,1,140,17,1,189,12
9
6020 DATA3,77,12,1,208,9,238,22,1,140,18,1,140,19,1,189,130,3,77,11,1,208,9,238
6030 DATA22,1,140,20,1,140,21,1,189,131,3,77,22,1,240,11,76,26,5,0,80,5,2,0,143
6040 DATA34,66,189,128,3,77,12,1,208,17,173,17,1,45,18,1,240,9,238,23,1,140,17,1
6050 DATA140,18,1,189,128,3,77,11,1,208,14,173,17,1,45,20,1,240,6,238,23,1,140,2
0
6060 DATA1,189,129,3,77,13,1,208,17,173,16,1,45,19,1,240,9,238,23,1,140,16,1,140
6070 DATA19,1,189,129,3,77,11,1,208,11,173,19,1,45,20,1,240,3,238,23,1,189,130,3
6080 DATA77,13,1,208,14,173,16,1,45,21,1,240,6,238,23,1,140,21,1,189,130,3,77,12
6090 DATA1,208,11,173,18,1,45,21,1,240,3,238,23,1,189,132,3,77,23,1,208,6,236,29
6100 DATA1,208,46,96,173,11,1,201,9,240,6,238,11,1,76,47,4,205,12,1,240,9,238,12
6110 DATA1,152,141,11,1,240,239,205,13,1,240,9,236,13,1,152,141,12,1,240,237,140
6120 DATA15,1,96,232,232,232,232,232,76,50,4,0

```

Mastermind 2

Bei dieser Version spielen zwei Spieler gegeneinander und versuchen zur glei-

chen Zeit die versteckte Zeichenkombination zu erraten, wobei jeder die

Versuche des Gegners und die eigenen in seine Strategie zur Lösungsfindung mit einbeziehen kann.

Die Spielanleitung am Anfang erklärt, wie das Spiel bedient wird. Hat man sei-

nen Versuch auf dem Bildschirm mittels Zahlen und den 'Größer-als- Kleiner-als-Tasten' plazierte, so schließt man ihn mit der Spacetaste ab.

```

1 H=2:GOSUB1000
5 INPUT"HWIEVIEL ZEICHEN ( 3 - 10 )":WZ:WZ=INT(WZ):IFWZ>10ORWZ<3THENS
10 DIMTR$(2,12,5),RP(2,12),RZ(2,12),CZ$(5),P$(14)
20 FORI=1TO5:CZ(I)=INT(RND(1)*WZ):NEXT
21 FORI=1TO5:IFCZ(I)=0THENCZ$(I)="#"
22 IFCZ(I)=1THENCZ$(I)="♦"
23 IFCZ(I)=2THENCZ$(I)="♣"
24 IFCZ(I)=3THENCZ$(I)="*"
25 IFCZ(I)=4THENCZ$(I)="♠"
26 IFCZ(I)=5THENCZ$(I)="♠"
27 IFCZ(I)=6THENCZ$(I)="♠"
28 IFCZ(I)=7THENCZ$(I)="♠"
29 IFCZ(I)=8THENCZ$(I)="♠"
30 IFCZ(I)=9THENCZ$(I)="♠"
31 NEXT
32 FORI=1TO14:READP$(I):NEXT
35 PRINT" "
50 F=0:IFH=1THENH=2:GOTO70
60 IFH=2THENH=1
62 DATA" ", " ", "0 = #", "1 = ♦", "2 = ♣", "3 = *", "4 = ♠", "5 = ♠", "6 = ♠"
63 DATA"7 = ♠", "8 = ♠", "9 = ♠"
64 DATA" ", " "
70 PRINT" ** MASTERMIND ** "
75 IFV<1THENFORI=1TO14:PRINT" ";P$(I);" ";NEXT
80 PRINT" "
82 PRINT" RETURN "
84 PRINT" "
95 PRINT" "
100 FORI=1TO12:FORJ=1TO5
110 PRINT" ";LEFT$(TR$(H,I,J),1);IFTR$(H,I,J)=" "THENPRINT" ";";
115 NEXT
120 PRINT" ";IFRP(H,I)=0THENPRINT" ";":LF=3
121 IFRP(H,I)=1THENPRINT" ";":LF=4
122 IFRP(H,I)=2THENPRINT" ";":LF=3
123 IFRP(H,I)=3THENPRINT" ";":LF=2
124 IFRP(H,I)=4THENPRINT" ";":LF=1
125 IFRP(H,I)=5THENPRINT" ";";
126 FORI1=1TOLF:PRINT" ";":NEXT
130 IFRZ(H,I)=1THENPRINT" ";";
131 IFRZ(H,I)=2THENPRINT" ";";
132 IFRZ(H,I)=3THENPRINT" ";";
133 IFRZ(H,I)=4THENPRINT" ";";
134 IFRZ(H,I)=5THENPRINT" ";";
140 PRINT
150 NEXT:IFF=1THEN50
200 F=1:L=L+1:V=INT(L/2.000001)+1
210 PRINT" SPIELER";H;" "
215 FORI2=0TO4:POKE32010+V*40+I2*3,191:NEXT
216 GOSUB600
221 GOSUB501
225 FORI=1TO5:CZ$(I)=LEFT$(CZ$(I),1):NEXT
230 FORI=1TO5:IFTR$(H,V,I)=CZ$(I)THENRP(H,V)=RP(H,V)+1:GOSUB900
235 NEXT
240 FORI=1TO5:FORJ=1TO5:IFTR$(H,V,I)=CZ$(J)THENRZ(H,V)=RZ(H,V)+1:GOSUB910
245 NEXTJ,I
250 IFRP(H,V)=5THEN300
260 IFV=12ANDH=2THEN400
270 GOTO50
300 PRINT" SPIELER";H;" GEWONNEN IM";V;" VERSUCH ";
310 GOTO430
400 PRINT" "
401 PRINT" ALLE SPIELER SIND NACH 12 VERSUCHEN GESCHEITERT!";
402 PRINT" "

```



```
405 PRINT"§
410 PRINT"§DER ZAHLENCODE LAUTETE: ";FORI=1TO5:PRINTLEFT$(CZ$(I),1);" ";NEXT:
PRINT
420 PRINT"§
430 INPUT"§NEUES SPIEL";JN$:IFLEFT$(JN$,1)="J"THENRUN
440 PRINT"§OKAY....."
445 POKE158,0:WAIT158,1:POKE158,0
450 END
501 FORI=1TO5:IFTR$(H,V,I)="0"THENTR$(H,V,I)="*"
502 IFCZ(I)=1THENCZ$(I)="♦"
503 IFCZ(I)=2THENCZ$(I)="♣"
504 IFCZ(I)=3THENCZ$(I)="♥"
505 IFCZ(I)=4THENCZ$(I)="♠"
506 IFCZ(I)=5THENCZ$(I)="♦"
507 IFCZ(I)=6THENCZ$(I)="♠"
508 IFCZ(I)=7THENCZ$(I)="♣"
509 IFCZ(I)=8THENCZ$(I)="♥"
510 IFCZ(I)=9THENCZ$(I)="♦"
511 NEXT
512 RETURN
600 I3=32810+V#40:I4=I3-1:POKEI4,158:POKE158,0
610 GETA$:IFA$=" "ANDPEEK(I4)=191THENPOKEI3,63
612 IFA$=" "ANDPEEK(I4)=63THEN610
614 IFA$="C"ANDI4>I3-1THENI4=I4-3:POKEI4,158:POKEI4+3,160
616 IFA$=">"ANDI4<I3+11THENI4=I4+3:POKEI4,158:POKEI4-3,160
618 IFA$=" "THENPRINT"§"
0700
620 IFA$="0"THENPOKEI4+1,163
625 IFA$="1"THENPOKEI4+1,211
630 IFA$="2"THENPOKEI4+1,193
635 IFA$="3"THENPOKEI4+1,170
640 IFA$="4"THENPOKEI4+1,216
645 IFA$="5"THENPOKEI4+1,218
650 IFA$="6"THENPOKEI4+1,233
655 IFA$="7"THENPOKEI4+1,255
660 IFA$="8"THENPOKEI4+1,164
665 IFA$="9"THENPOKEI4+1,230
670 GOTO610
700 I6=0:FORI5=I3TO(I3+12)STEP3:I6=I6+1
701 IFPEEK(I5)=211THENTR$(H,V,I6)="♦"
702 IFPEEK(I5)=193THENTR$(H,V,I6)="♣"
703 IFPEEK(I5)=170THENTR$(H,V,I6)="♥"
704 IFPEEK(I5)=216THENTR$(H,V,I6)="♠"
705 IFPEEK(I5)=218THENTR$(H,V,I6)="♦"
706 IFPEEK(I5)=233THENTR$(H,V,I6)="♠"
707 IFPEEK(I5)=255THENTR$(H,V,I6)="♣"
708 IFPEEK(I5)=164THENTR$(H,V,I6)="♥"
709 IFPEEK(I5)=230THENTR$(H,V,I6)="♣"
710 IFPEEK(I5)=163THENTR$(H,V,I6)="*"
712 NEXT
720 FORI5=I3-1TOI3+11STEP3:POKEI5,160:NEXT
730 RETURN
900 TR$(H,V,I)=TR$(H,V,I)+"0":CZ$(I)=CZ$(I)+"0":RETURN
910 TR$(H,V,I)=TR$(H,V,I)+"0":CZ$(J)=CZ$(J)+"0":RETURN
1000 PRINT"§"
1010 PRINT"§MASTERMIND #2 "
1040 PRINT"§FINDEN SIE DEN VERSTECKTEN ZEICHENCODE."
1050 PRINT"§MIT DEN TASTEN § < § & § > § HABEN SIE"
1055 PRINT"§DIE MOEGlichkeit DEN PLATZ ZU BESTIMMEN,"
1060 PRINT"§AN DEN SIE EIN BESTIMMTES ZEICHEN SET-"
1065 PRINT"§ZEN MOECHTEN ( EINGABE EINER ZAHLI >."
1070 PRINT"§MIT DER LEERTASTE BEENDEN SIE IHREN VER-"
1075 PRINT"§SUCH."
1080 PRINT"§DIE §-ZEICHEN ZEIGEN AN: CODEZEICHEN ISTAN DER RICHTIGEN STELLE!"
1085 PRINT"§DIE §-ZEICHEN ZEIGEN AN: CODEZEICHEN ISTRICHTIG ABER AN DER FALSCHEN
N";
1090 PRINT" §STELLE!"
1100 RETURN
```

":GOT

Der VC-64 Videochip Teil 1

In diesem Heft beginnt eine Serie, in der dem Computer etwas näher in die Bytes geschaut werden soll.

Die maximale Anzahl der Zeilen im Tastaturpuffer läßt sich durch POKE 649, N bestimmen, wobei N eine Integerzahl zwischen 0 und 255 ist. Tippt man noch mehr Tastendrucke ein bevor die nächste GET Anweisung kommt, so werden die weiteren Tastendrucke ignoriert. Es ist jedoch nicht empfehlenswert, in 649 eine größere Zahl als 10 zu poken, da der 11te abgespeicherte Tastendruck den Computer aussteigen läßt. Man kann das Basic umändern und sich sozusagen

sein eigenes Basic schreiben. Dazu muß man erst das Basic aus dem ROM in den RAM-Bereich gleicher Adresse kopieren. Das macht man mit:

```
FOR I = 40960 TO 49151:
POKEI, PEEK (I): NEXT.
Jetzt muß man dem Computer noch sagen, daß das Basic im RAM steht. Das tut man mit POKE 1,54. Zurückschalten kann man mit POKE 1,55.
```

Nun kann man nachsehen, an welcher Stelle die einzelnen Befehlswoorte stehen; z. B. indem man die STOP-Taste drückt, wenn

das gesuchte Wort kommt und dann die Variable I drückt. Dann PEEKed man die Bytes in der Nähe von I ab, um den genauen Wert zu bekommen. Per POKE kann man jetzt die Befehlswoorte umändern. Man muß aber darauf achten, daß man bei dem letzten Buchstaben oder Zeichen das höchstwertige Bit setzt, d. h. zu dem ASCII-Code 128 addiert, und bei den anderen nicht, da sich sonst die Interpretercodes der darauffolgenden Befehle und Funktionen ändern. So kann man z. B. mit POKE 41118, ASC ("U") den Befehl END in UND umändern. Dann kann man schreiben: END = 7 : PRINTEND und erhält die 7. Dagegen führt die Anweisung END zu einem ?SYNTAX ERROR. Am Programmende kann man UND statt END schreiben.

Die Befehlswoorte stehen ab 41118 bis 41372 in der Reihenfolge ihres Interpreter-

tercodes. Der erste Befehl, END, hat den Interpretercode 128. Das Ende eines Befehlswoortes ist dadurch gekennzeichnet, daß das höchstwertige Bit gesetzt ist.

Wenn man die Befehlswoorte auslisten will, muß man das höchstwertige Bit löschen, da sonst statt des letzten Buchstabens des Befehlswoortes das Graphikzeichen erscheint, das man erhält, wenn man den letzten Buchstaben mit gedrückter Shift-Taste tippt. Hier ein Programm dazu:

```
10 FOR I = 41118 TO 41372
20 P = PEEK(I):
PRINTCHR$ (127 AND
P); IF P > 127 THEN
PRINT
30 NEXT
```

Im nächsten Heft werden wir uns dann näher mit dem VC-64-Zeichensatz und seine Möglichkeiten befassen.

Homecomputer

*Sonderheft
Hannover Messe*



**Hannover
Messe '83**

ab 20. Mai '83

bei Ihrem Zeitschriftenhändler

oder direkt bestellen beim:

Roeske Verlag

Postfach 629

3440 Eschwege

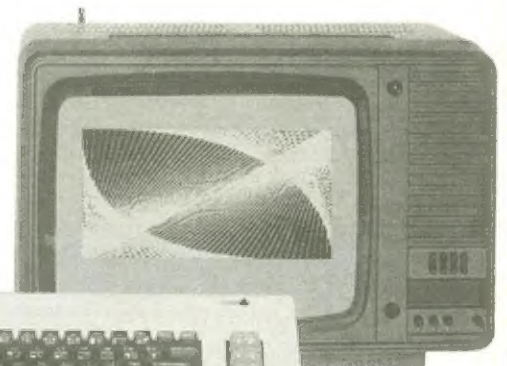
250 Computer zu gewinnen.

Jeder Abonnent, der einen Abonnenten für Homecomputer wirbt, erhält als kleines Dankeschön "Das Microcomputer-Taschenbuch" (Erscheinungstermin Juni 1983) mit viel Wissenswertem über die populärsten Microcomputer. Außerdem verteilen wir noch 250 Computer an diejenigen Leser, die Homecomputer die meisten Abonnenten zuführen.

Die Gewinnverteilung erfolgt nach einem für alle Leser sehr interessanten Schlüssel. Jedesmal, wenn bei uns 1000 neue Abonnenten registriert wurden, dann erhalten diejenigen 10 Leser, die zu diesem Zeitpunkt die meisten Abonnenten geworben haben – also in der Wettbewerbsliste vorne liegen – einen Microcomputer ihrer Wahl und zwar der Erste dieser Wettbewerbsliste einen Microcomputer im Werte von 2500,- DM und die 9 folgenden jeder einen im Werte von 1000,- DM.

Die 10 Gewinner werden aus der Wettbewerbsliste gestrichen (sie können allerdings von neuem beginnen und wieder gewinnen), und die ihnen Folgenden rücken nach.

Sind wieder 1000 Abonnenten erreicht, geht es weiter wie oben beschrieben.



Teilnahmeberechtigt ist jeder, der Homecomputer regelmäßig im Abonnement bezieht.

Die Gewinner werden sofort benachrichtigt und in der jeweils folgenden Ausgabe von Homecomputer bekanntgegeben.

Leser-Software-Service Korrekturen

K = Kassette D = Diskette

TI-99/4A K 10,- DM

Panzerkrieg

TI-Ufo

Killersatellit

Chiffrier/dechiffrier-PGM

Rangliste

VC-20 K 14,- DM

D 20,- DM

Invaders (3PGME)

Editor

Assembler (neu)

Labyrinth

Car-Crash

ZX-81 K 8,- DM

Ritter

Missile ZX-Command

VC-64 K 8,- DM

D 15,- DM

Mondlandung

Wurm

Geisterjäger

VC-64-Sprite-Generator

Apple II D 12,- DM

Diskschutz

Trollhöhlen

Pet-Bytes K 8,- DM

Mastermind 1

Mastermind 2

Programme auf Diskette und Kassette

Nicht jeder Leser hat die Zeit und die Geduld, alle Programme, die er gerne hätte, in sein Gerät einzutasten.

Wir haben uns deshalb überlegt, daß es das einfachste ist, wenn wir die Programme, die wir ohnehin auf Diskette oder Kassette vorliegen haben, diesem Leserkreis zum Selbstkostenpreis (Datenträger, Kopierzeitaufwand, Porto, Verpackung und Mehrwertsteuer) anbieten.

Es genügt jeweils die Angabe eines Stichwortes, z. B.: VC-20-Disk Heft 5 oder Sinclair-Kassette Heft 5.

Bestellungen Inland:

Gegen Einsendung eines Schecks oder Vorauszahlung auf unser Konto bei der Kreissparkasse Eschwege, Bankleitzahl 522 500 30

Nummer 45 22 934 senden wir Ihnen die gewünschten Programme schnellstmöglich zu.

Bestellungen Ausland:

Nur Vorkasse Schein (Kassette 10,- DM, Diskette 20,- DM). Keine Schecks oder Überweisungen!

Lieferung noch nicht erhalten?

Bei Überweisung auf unser Konto kann es bis zu 2 Wochen dauern, bis wir Ihre Bestellung in Händen haben

Oft passiert es, daß auf der Überweisungsdurchschrift weder Name, noch Ort, noch Art der Bestellung zu erkennen sind. Schreiben Sie uns! (Anrufe kosten viel Geld und bringen, weil dann Schriftvergleiche nicht möglich sind, kein Ergebnis!)

Heft 5/83

Spät, aber nicht zu spät fiel auf, daß das Listing vom **Assembler** in diesem Heft die „Generalproben-Version“ war.

Später wurden noch 3 Zeilen geändert

```
1070 IF AI < 6 AND L > 1
THEN . . .
4135 AI = 0: (Rest wie abgedruckt)
10000 DATA (der Wert 170 muß durch 176 ersetzt werden)
```

Heft 4/83

Das Listing des Programmes **Asteroiden** hatte folgenden Fehler:

```
930 IF KEY=74 THEN 170
```

Auftrag für Gelegenheitsanzeigen in Homecomputer

An Homecomputer
Gartenstraße 47
3440 Eschwege

Name und Adresse _____

Abo.-Nr. _____

Unterschrift _____

Ich zahle sofort nach Rechnungserhalt.

Datum _____

Unter der Rubrik „Kleinanzeigen“ veröffentlichen wir Gelegenheitsanzeigen für Verkaufsangebote, Kauf- und Tauschgesuche, Kontaktaufnahme bzw. Erfahrungsaustausch usw.

Preise für „Kleinanzeigen“:
Private Gelegenheitsanzeige je Druckzeile 5,- DM inkl. MwSt. Chiffregebühr je Anzeige 10,- DM.

Gewerbliche Gelegenheitsanzeige je Druckzeile 7,- DM inkl. MwSt. (dürfen nicht unter Chiffre erscheinen).

Ich wünsche folgenden Text zu veröffentlichen:

Zutreffendes ankreuzen

suche Software

suche Hardware

Tausch

Versch.

biete an Software

biete an Hardware

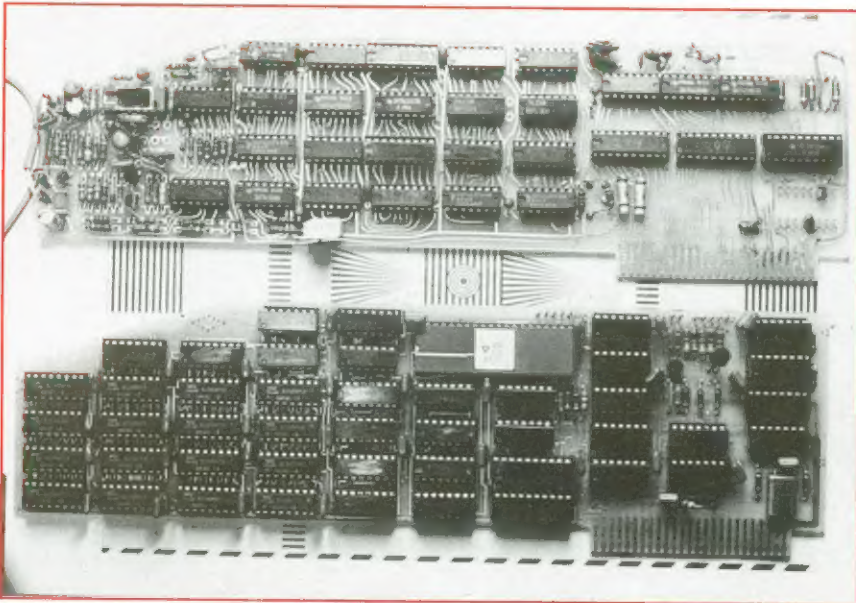
Kontakte

Chiffre

BARTSCHER ELEKTRONIK

Rüsten Sie Ihren Apple auf mit unseren Video-Interfaces!

(auch für Apple kompatible Computer)



AD 16-3

Echtzeitdigitalisierer

1356,- DM

incl. MWSt.



HGR-512

Graphicinterface

1356,- DM

incl. MWSt.

HGR-512 Graphicinterface

- Eigener 32 k Byte Bildspeicher
- Graphic-Controller EF 9365
- Umschaltbar 512 x 512 Punkte in schwarz-weiß oder 256 x 256 in 16 Graustufen bzw. 16 Farben auf jedem dieser Punkte
- Hardcopymöglichkeit von 512er-Modus
- ASCII-Zeichensatz in Größe und Richtung programmierbar
- Vektorplot mit 1 Millionen Punkte pro Sekunde
- Ausgang für RGB- und BAS-Monitore
- Demosoftware auf Diskette

AD 16-3 Echtzeitdigitalisierer

- Echtzeitdigitalisierer in 40 m sec.
- ca. 65 000 Bildpunkte mit 16 Graustufen
- eingebauter DA-Wandler zur Betrachtung des digitalisierten Bildes
- Digitalisierung von Videorecorder, Kamera oder Fernseher
- Keine Software für Bildlesevorgang nötig

NEUANKÜNDIGUNG:

AD 16-4

Echtzeitdigitizer mit
Sofortdisplay und
eigenem Bildspeicher

Digitalisierung und Display geschieht in 1/50 sec.
Karte enthält eigenen Bildspeicher.
Auflösung: 256 x 256 Bildpunkte.
Fremdsynchronisation der Kamera notwendig.



mit AD16-3 digitalisiert durch HGR-512 dargestellt

Bartscher Elektronik · Westring 84 · 3440 Eschwege · Tel.: 0 56 51/17 95

WICOSOFT

Microcomputer
SOFTWARE

WICO-SOFT

Programm-
Zusendungen
erwünscht!

Neu:

Demnächst
Software für alle
Systeme lieferbar!

Der Fluch des Pharao

Adventure-Spiel f. VC-20

(Speichererweiterung mind. 16K)

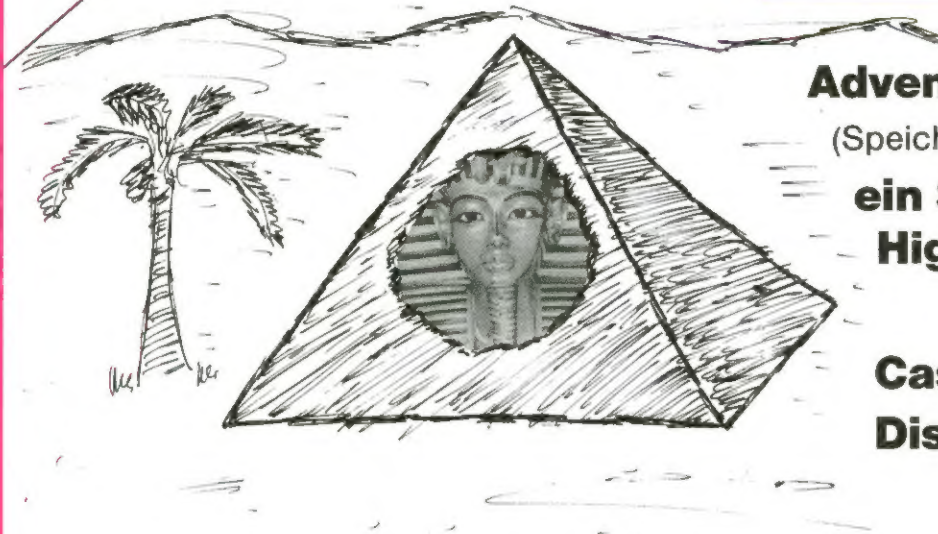
**ein Superspiel m.
High-Res.-Graphik**

Cassette

19.50 DM

Diskette

25.00 DM



5 Programme für den VC-20 auf einer Cassette!
Geschicklichkeits- und Denkspiele f. Joystick und
Tastenbedienung.

- Autorennen
- Rememby
- Roulette
- Mastermind
- Fadenkreuz

alles zusammen nur

19.50 DM

- V-Scheck ist beigelegt
- per Nachnahme
(zzgl. Nachnahmegebühr)

WICOSOFT

Nordstr. 22, 3443 Herleshausen

Ich bestelle:

Der Fluch des Pharao

VC-Programmkassette

Cassette 19.50 DM

Diskette 25.00 DM

VC-20 o.Erw.

VC-20 +8K

Name, Anschrift: _____